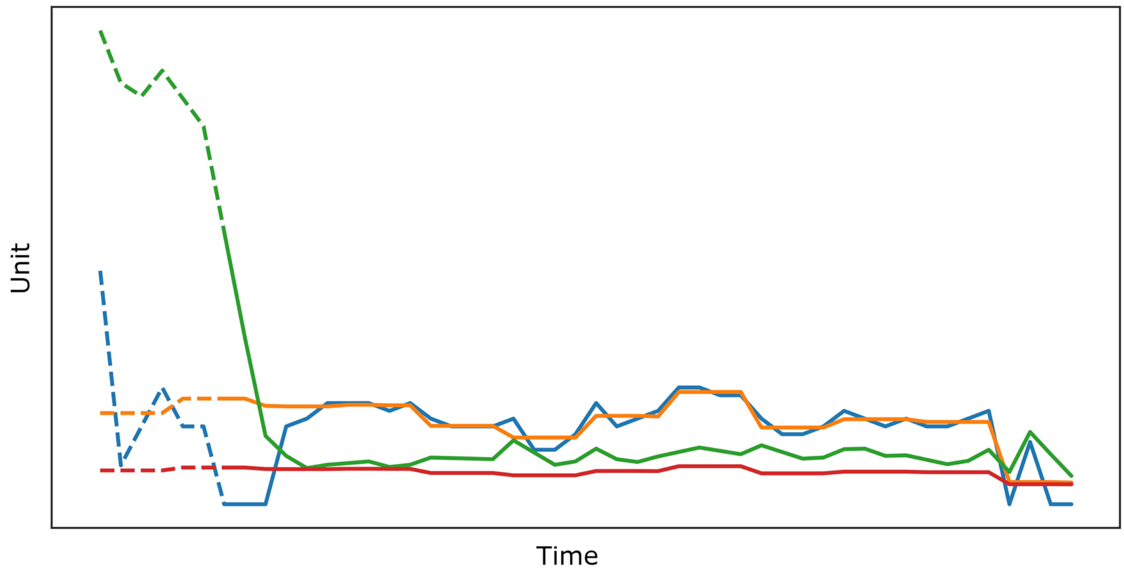
JYU DISSERTATIONS 657

Adhe Kania

Addressing Challenges of Real-World Lot Sizing Problems with Interactive Multiobjective Optimization



UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY

JYU DISSERTATIONS 657

Adhe Kania

Addressing Challenges of Real-World Lot Sizing Problems with Interactive Multiobjective Optimization

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi Agoran auditoriossa 2
kesäkuun 8. päivänä 2023 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 2, on June 8, 2023, at 12 o'clock.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2023

Editors
Marja-Leena Rantalainen
Faculty of Information Technology, University of Jyväskylä
Päivi Vuorio
Open Science Centre, University of Jyväskylä

ISBN 978-951-39-9637-6 (PDF)
URN:ISBN:978-951-39-9637-6
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-9637-6

ABSTRACT

Kania, Adhe
Addressing challenges of real-world lot sizing problems with interactive multi-objective optimization
Jyväskylä: University of Jyväskylä, 2023, 70 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 657)
ISBN 978-951-39-9637-6 (PDF)

Many real-world problems involve multiple conflicting objective functions to be optimized simultaneously, including lot sizing problems, where we need to minimize costs while satisfying demand. A multiobjective optimization problem has many so-called Pareto optimal solutions reflecting different trade-offs. A decision maker (DM) is needed to select one of them to be applied in practice representing best his/her preferences. Interactive methods, that iteratively incorporate the DM's preferences, are beneficial in supporting the DM, and therefore, in this thesis, we focus on solving lot sizing problems with interactive methods.

This thesis tackles challenges in modeling and solving lot sizing problems inspired by real challenges. First, we consider a single-item lot sizing problem under demand uncertainty and propose a safety order time concept that can efficiently handle high fluctuations on demand. Second, we focus on a single-item lot sizing problem under demand and lead time uncertainties, and propose a probability of product availability formula to assess the quality of safety lead time. Third, we integrate a lot sizing problem and a minimum order quantity (MOQ) determination and propose a MOQ level formula to measure the quality of MOQ in satisfying demand. Besides, we also propose multiobjective optimization models to solve these problems. Last, we address a challenge in multi-item lot sizing problems by proposing a decision support approach, called DESMILS. DESMILS enables any single-item multiobjective lot sizing models to be applied in solving multi-item problems by accommodating different preferences from the DM.

As a proof of concept, we utilized real data from a company to demonstrate the applicability of the proposed models and approaches. We supported the supply chain manager of the company, as the DM, to find his most preferred solutions by solving the proposed single-item lot sizing models, with interactive methods or the hybridization of methods that we propose. We then demonstrate that, with DESMILS, the DM found Pareto optimal lot sizes for 94 items by solving a single-item multiobjective lot sizing problem for only ten representative items. The DM found all concepts, models, interactive decision making processes, and results useful in his daily operations. These successful applications demonstrate the practical value of the research, which can also benefit others in lot sizing.

Keywords: Inventory management, multi-item, demand uncertainty, lead time uncertainty, minimum order quantity, decision support

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Kania, Adhe
Ratkaisuja todellisten toimituserän mitoitusongelmien haasteisiin interaktiivisen monitavoiteoptimoinnin avulla
Jyväskylä: University of Jyväskylä, 2023, 70 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 657)
ISBN 978-951-39-9637-6 (PDF)

Monissa tosielämän ongelmissa on useita optimoitavia tavoitefunktioita, jotka ovat ristiriidassa keskenään. Esimerkiksi tuotantoyritysten varaston ohjauksessa on toimituseriä mitoitettaessa minimoitava kustannuksia ja samalla varmistettava nimikkeiden riittävyys sekä tarpeiden mukainen varastotaso. Monitavoiteoptimointiongelmilla on monia ns. Pareto-optimaalisia ratkaisuja, jotka kuvastavat tavoitefunktioiden välisiä vaihtosuhteita. Tarvitaan päätöksentekijä valitsemaan yksi niistä käytäntöön vietäväksi päätökseksi, joka parhaiten kuvasta hänen mieltymyksiään. Iteratiivisesti mieltymykset huomioon ottavat interaktiiviset menetelmät tukevat päätöksentekijää tehokkaasti. Siksi tässä väitöskirjassa tuetaan interaktiivisten menetelmien avulla toimituserän kokoon ja ajoitukseen liittyvää päätöksentekoa varaston ohjauksessa.

Väitöskirja mallintaa ja ratkaisee käytännön haasteista nousevia toimituserän mitoitusongelmia. Ensin yhden nimikkeen toimituserää optimoidaan kysynnän vaihdellessa ja esitellään uusi käsite, varmuusaika. Sitten yhden nimikkeen toimituserää optimoidaan, kun kysyntää on vaikea hallita ja toimitusajat ovat epävarmat. Esiteltävä uusi kaava määrittää varmuusajan nimikkeen riittävyyden todennäköisyydelle. Kolmanneksi yhdistetään toimituserän mitoitus ja minimitoimituserän määrittely. Kaikkien näiden kolmen haasteen ratkaisemiseksi muotoillaan monitavoiteoptimointiongelmat ja ratkaistaan ne. Lopuksi esitellään uusi päätöksenteon tukimenetelmä DESMILS usean nimikkeen toimituserän optimointiin. Sen avulla mitä tahansa yhden nimikkeen monitavoitteinen toimituserän mitoitusmalli voidaan tehokkaasti laajentaa usealle nimikkeelle päätöksentekijän mieltymykset huomioiden.

Mallien ja menetelmien soveltuvuutta havainnollistetaan tuotantoyrityksen datalla. Päätöksentekijänä toiminutta yrityksen toimitusketjun johtajaa tuettiin löytämään parhaat ratkaisut eri ongelmiin käyttäen interaktiivisia menetelmiä tai niiden yhdistelmiä. Lisäksi DESMILS auttoi päätöksentekijää mitoittamaan Pareto-optimaaliset toimituserät 94 nimikkeelle niin, että hänen täytyi mitoittaa vain 10 huolella valitun nimikkeen tilausmäärät. Päätöksentekijästä kaikki käsitteet, mallit, interaktiiviset ratkaisuprosessit ja ratkaisut olivat hyödyllisiä ja tukivat päivittäisiä toimintoja. Onnistuneet tulokset havainnollistavat tutkimuksen käytännön arvoa ja hyötyä myös muiden toimituserän optimointiongelmien ratkaisemisessa.

Avainsanat: varaston hallinta, vaihtelevat toimituserät, kysynnän epävarmuus, toimitusajan epävarmuus, minimitoimituserä, päätöksenteon tuki

Author Adhe Kania
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors Professor Kaisa Miettinen
Faculty of Information Technology
University of Jyväskylä
Finland

Dr. Bekir Afsar
Faculty of Information Technology
University of Jyväskylä
Finland

Reviewers Professor Martin Josef Geiger
Helmut-Schmidt University
University of the Federal Armed Forces
Germany

Associate Prof. Jafar Rezaei
Faculty of Technology, Policy and Management
Delft University of Technology
The Netherlands

Opponent Professor Amos H. C. Ng
School of Engineering Science
University of Skövde
Sweden

ACKNOWLEDGEMENTS

First, I would like to express my deepest gratitude to my supervisors, Prof. Kaisa Miettinen and Dr. Bekir Afsar, for their continuous support, guidance, motivation, and insightful advice throughout my doctoral studies. Their expertise and mentorship have been instrumental in shaping the direction and quality of my research. I am truly grateful for the opportunity to work under their supervision.

Besides my supervisors, I would also like to extend my sincere thanks to Mr. Juha Sipilä and Mr. Jussi Lehtimäki, as experts in lot sizing problems and co-authors in my articles, for their valuable discussions, sharing, and thoughtful feedback throughout my research journey. Their expertise and input have greatly enriched the quality of this work. For reviewing this thesis, I would like to thank to Prof. Martin Josef Geiger from Helmut-Schmidt University and Assoc. Prof. Jafar Rezaei from Delft University of Technology for providing valuable insights and constructive feedback on my thesis.

Furthermore, I would like to express my appreciation to the Indonesian Endowment Fund for Education (LPDP) and the Faculty of Information Technology at the University of Jyväskylä for their generous financial support. Their funding has been instrumental in facilitating the execution of this research, enabling me to pursue my academic goals. I would also like to acknowledge all the current and previous members of the Multiobjective Optimization Group who provided a conducive research environment through discussions and seminars. Especially, I want to thank Giovanni Misitano for his support related to the user interface and being my co-author. Also, thanks to Dr. Bhupinder S. Saini for his help and shared experiences related to DESDEO and Python programming.

I would like to express my heartfelt thanks to my family in Indonesia, especially for my parents, Mr. Harun Rosidi and Mrs. Siti Anisah, for their tireless support, motivation, prayers, and hopes. I would also like to thank warmly my dear and loyal Indonesian friends in Jyväskylä who have been my family in Finland. Especially for Rita, kak Suzi, Evie, and their family, for brightening my daily life in Finland. Lastly, but certainly not least, I extend my deepest gratitude to my beloved family: my husband Muhammad Subair and my children Shafiyyah Kalila, Rumaisha Hanifah, and Ibrahim Athaya. Their unwavering support, understanding, and patience have been a constant source of motivation and inspiration throughout this endeavor. I am truly grateful for their love, encouragement, and sacrifices, which have enabled me to pursue my academic aspirations.

In conclusion, I am humbled and grateful for the contributions and support of each and every individual who has played a part in my journey, but cannot be mentioned one by one. Most of all, I thank Allah SWT for all the blessings bestowed upon me, shaping me into the person I am today. Alhamdulillahi rabbil 'alamin.

LIST OF ACRONYMS

CSL	Cycle service level
DM	Decision maker
HC	Holding cost
ITO	Inventory turnover
ML	MOQ Level
MOQ	Minimun order quantity
OC	Ordering cost
PC	Purchasing cost
POC	Purchasing and ordering cost
PPA	Probability of product availability
SLT	Safety lead time
SOT	Safety order time
SS	Safety stock

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Adhe Kania, Juha Sipilä, Giovanni Misitano, Kaisa Miettinen, and Jussi Lehtimäki. Integration of lot sizing and safety strategy placement using interactive multiobjective optimization. *Computers & Industrial Engineering, vol 173, article no. 108731*, 2022.

PII Adhe Kania, Juha Sipilä, Bekir Afsar, and Kaisa Miettinen. Interactive multiobjective optimization in lot sizing with safety stock and safety lead time. In: Mes, M., Lalla-Ruiz, E., Voß, S. (eds), Computational Logistics. 12th International Conference, ICCL 2021, Proceedings, pp. 208–221, 2021.

PIII Adhe Kania, Bekir Afsar, Kaisa Miettinen, and Juha Sipilä. Determining minimum order quantity and lot sizes from the buyer's perspective with interactive multiobjective optimization. *Submitted to a journal*.

PIV Adhe Kania, Bekir Afsar, Kaisa Miettinen, Juha Sipilä. DESMILS: A decision support approach for multi-item lot sizing using interactive multiobjective optimization. *Journal of Intelligent Manufacturing (to appear)*, https://doi.org/10.1007/s10845-023-02112-5, 2023.

1 INTRODUCTION

In today's highly competitive market, many companies face challenges in managing their production and inventory efficiently. Lot sizing plays a significant role to improve efficiency in inventory management. The purpose of lot sizing is to determine the optimal quantity of item(s) to be ordered to meet the needs of manufacturing in the period(s) considered and minimizing costs at the same time. Economic order quantity [25] is the first model proposed to solve a simple lot sizing problem. After this invention, numerous variants and extensions of lot sizing models have been proposed in the literature (see e.g. the surveys [6, 19]).

Minimizing costs while satisfying demand in lot sizing problems naturally introduces conflicting objective functions. Nevertheless, this problem is often modeled as a single objective model. For example, some studies combine all objective functions into a single objective function as a weighted sum, while others select one of the objective functions to be optimized and set others as constraints [13]. In this way, information regarding interdependencies among the objective functions may be lost and this may affect the validity of the solutions obtained. Even though a solution is an optimal solution for the single objective problem, it may not be the best possible solution for the original problem. Therefore, the problem with multiple conflicting objective functions should be treated as a multiobjective optimization model (see e.g. [36]), where all objective functions are considered simultaneously. Some studies have utilized multiobjective optimization to address various topics in lot sizing [7], such as supplier selection [48], perishability issues [4], and sustainability concerns [8]. This thesis studies the benefits that multiobjective optimization can offer in solving real-world lot sizing problems.

In optimization problems with multiple conflicting objective functions, identifying a single optimal solution that optimizes all objective functions is typically impossible. Instead, a multiobjective optimization problem has many solutions, called Pareto optimal solutions, at which an improvement in one objective function value is only possible at the expense of impairment in the value(s) of, at least, one of the others. Without any additional information, these solutions are equally good and mathematically incomparable, but one solution needs to be selected among them to be applied in practice. This places a significant responsibility on a decision maker (DM), who is an expert in the problem domain, to provide preference information used to find his/her most preferred solution as a final one. This means that the final solution of a multiobjective optimization problem is the most preferred (Pareto optimal) solution for the DM.

Finding the most preferred solution among many alternative solutions can be a challenging task for the DM without any support. Therefore, methods that offer support to the DM to learn about the trade-offs among the objective functions and adapt his/her preferences while learning, can increase the confidence and satisfaction of the DM with the final solution. Interactive methods (see e.g. [38, 42]), which iteratively incorporate the DM's preferences during the decision making process, provide such support. These methods allow the DM to take an active role in the decision making process and gain a better understanding of the trade-offs among different objective functions. By iteratively adjusting his/her preferences and exploring various solutions, the DM can gain insights into the problem and make more informed decisions. As a result, interactive methods can improve the quality of the final solution and increase the possibility of its successful application. Over the years, many interactive methods have been developed [36, 38] and have provided promising solutions in various fields of application. However, only few studies have applied interactive methods in lot sizing problems [27]. For these reasons, we concentrate on interactive methods in this thesis.

When utilizing interactive methods, we often observe two phases on the DM: a learning phase and a decision phase [42]. In the learning phase, the DM explores various solutions that reflect different preferences and increases his/her understanding of the problem until he/she identifies a region of interest. Then, in the decision phase, the DM refines the solutions within the region of interest until he/she finds his/her most preferred solution. Depending on the problem to be solved, a single method may be used for both phases, or some methods can be hybridized to combine their strengths in different phases to efficiently determine the most preferred solution for the DM.

Various challenges are often encountered when dealing with real-world multiobjective optimization problems [2]. One of them is modeling the problem accurately. The process of formulating the problem can be a complex task. The choice of objective functions to be optimized can depend on several factors, such as data availability, the needs of the stakeholders, or key performance indicators of the company. This formulation process can even evolve during the decision making process. Additionally, in modeling the problem, usually, not all formulas can be found in the literature, especially when tackling new challenges that have not been considered previously. Then, a novel formula also needs to be formulated.

Furthermore, finding a suitable method to support the DM in determining his/her most preferred solution for the model formulated can be challenging as well [1]. Even if an interactive method is to be applied, there are many methods available to choose from. Among them, we cannot say that one method is generally superior to all the others, but every method has its own strengths [42]. The selection of the method usually depends on the problem to be solved, as well as desire of the DM and his/her experiences in solving multiobjective optimization problems. Thus, careful consideration is necessary when selecting the appropriate objective functions and the method(s) to address the specific multiobjective optimization problem. Additionally, many real wold problems, including lot sizing problems, are known to be computationally expensive [3, 11]. This presents an additional challenge, since long computational times in solving these problems potentially cause significant delays that can frustrate the DM. Thus, it is important to minimize the waiting time of the DM in the decision making process in order to ensure that he/she can efficiently provide his/her preferences to find the final solution.

This thesis is a collection of four articles [PI]-[PIV], which are published or submitted in scientific journals and conference proceedings. Inspired by real challenges in lot sizing, we develop models for various real needs and then propose appropriate interactive multiobjective optimization methods to be applied to solve the corresponding problems. Specifically, we tackle the four following challenges:

C1 How do we solve a single-item lot sizing problem together with a strategy to handle uncertainty on demand efficiently?

C2 How do we solve a single-item lot sizing problem and simultaneously determine a safety stock and a safety lead time to handle demand and lead time uncertainty?

C3 How do we integrate a single-item lot sizing problem with a minimum order quantity determination problem?

C4 How can we use single-item lot sizing models to solve multi-item lot sizing problems that accommodate different preferences from the DM for different items without exhausting the DM with too many decision making processes?

Challenges C1-C3 are tackled in articles [PI]-[PIII], respectively, and discussed in Chapters 3 and 4. Challenge C4 is addressed in article [PIV] and introduced in Chapter 5.

In article [PI], we focus on lot sizing under demand uncertainty to tackle challenge C1. In this problem, a safety stock, as a traditional model to handle demand uncertainty (see e.g. [22, 23, 44]), cannot handle high fluctuations in demand [9], and the dynamic safety stock [28, 46] is unsuitable for problems with a large number of decision variables and constraints [57]. Therefore, we propose a safety order time concept that keeps additional stock based on time, together with the safety stock, to handle the stochasticity of demand efficiently. We also propose a multiobjective optimization model and modify the existing formulas to adapt the safety order time in the proposed model. In article [PII], we formulate a multiobjective optimization model to solve a lot sizing problem under demand and lead time uncertainty to tackle challenge C2. We use a safety lead time [59] to handle lead time uncertainty but lack the formula to measure the quality of safety lead time. To overcome this, we propose the probability of product availability formula and use it as one of the objective functions in the proposed model. Similarly, in article [PIII], when integrating lot sizing with the minimum order quantity determination problem to tackle challenge C3, there is no formula available in the literature to measure the quality of the minimum order quantity in satisfying demand. Hence, we propose a minimum order quantity level formula as well as a multiobjective optimization model to address this challenge. All the proposed models are described in Chapter 3.

This thesis is an instance of data-driven decision support, where multiobjective optimization is applied. Our proposed models are inspired by real challenges in lot sizing. We use real data as a proof of concept to demonstrate the applicability of the proposed models. Our motivation is to bridge the gap between theory and practice. All data used in this thesis is provided by a (manufacturing) company, and the supply chain manager of the company acted as the DM to provide his preferences in the decision making processes based on his expertise in real life. These case studies are presented in Chapter 4.

To solve the multiobjective optimization problems, which are formulated in the proposed models, as said, it is important to find suitable method(s) to support the DM to determine the most preferred solution for each of the problems defined. In articles [PI]-[PIII], we used variants of NAUTILUS methods [41]. These methods start from the worst possible objective function values and iteratively improve all objective functions based on preference information provided by the DM. In this way, the DM is able to find his/her most preferred solution without having to trade-off among the objective functions. This also allows the DM to avoid anchoring around the starting point, where he/she may be reluctant to move from the starting point due to the difficulty of sacrificing in at least one objective function to find a new Pareto optimal solution [14].

In solving a multiobjective optimization problem with a real DM, the DM must understand the meaning of each piece of information given to him/her in each iteration before providing his/her preferences. The DM in this thesis did not have any previous experience with multiobjective optimization in the beginning, and the case study in article [PI] was the first experience for the DM to conduct a supported decision making process. For this first experience, we suggested the DM use the E-NAUTILUS method [50], which is a variant of NAUTILUS methods, where in each iteration, the DM is presented with some solutions (with different objective function values) to be compared. In this way, the DM can easily understand the meaning of the information provided, and he gains an improvement in all the objective function values from iteration to iteration until a Pareto optimal solution is reached. We demonstrate in Chapter 4 that the DM successfully found his most preferred solution to solve challenge C1 with E-NAUTILUS.

After experiencing E-NAUTILUS, the DM understood more about multiobjective optimization, and we suggested using NAUTILUS Navigator [49] in article [PII], where the DM needs to provide a desirable value for each objective function at the beginning of the decision making process. This method then uses navigation ideas [26] to direct the movement, based on these desirable values, from the worst starting point to a Pareto optimal solution as the final solution. During the navigation process, the DM can navigate by changing his/her desirable values, the movement speed, or even go back, until he finds his most preferred solution. In Chapter 4, we also demonstrate how we support the DM to find the most preferred solution for him to solve challenge C2 with NAUTILUS Navigator.

In article [PIII], we propose a hybridization of methods that combines the strength of two interactive methods: NAUTILUS Navigator [49] and NIMBUS [40]. We use the synchronous NIMBUS method, as its type of providing preference information in terms of classification was regarded preferable by the DM. However, providing a good starting point for NIMBUS is important, as it greatly affects the final solution [14]. To achieve a good starting point for NIMBUS, we propose to use the NAUTILUS Navigator method. With this method, the DM can find a preferred Pareto optimal solution that can serve as a starting point for NIMBUS. In this way, NAUTILUS Navigator may support the DM in the learning phase to gain insights into the problem and enable navigation without trading-off until he/she obtains a solution representing the region of his/her interest. Then, NIMBUS supports him/her in the decision phase to refine the solution until his/her most preferred solution is obtained. We demonstrate in Chapter 4 that the DM appreciated the hybridization of methods that helped him to find his most preferred solution in an easier way to solve the problem addressing challenge C3.

In order to address the computational challenge of lot sizing problems, in all articles included in this thesis, we generated a large number of solutions that represented Pareto optimal solutions using an evolutionary method. The generation processes of these solutions, which can be time-consuming because of the expensive functions, were done before the interactive decision making processes, that involved the DM, were conducted. Then, the decision making processes were conducted by utilizing this representative set to reduce waiting times on the DM.

Many lot sizing studies only focus on a single item [13], yet in practice, companies must determine order quantities for numerous items. In this condition, repeating the decision making process for every single item is laborious. Furthermore, the few studies that focus on multi-item lot sizing problems treat each item similarly by combining the objective function(s) of all items into a sum. In this way, different preferences from the DM in lot sizing decisions for different items cannot be accommodated. This fact motivates us in challenge C4, to extend the single-item lot sizing model to be used to solve multi-item lot sizing problems in such a way that the DM is allowed to provide different preferences for different items. To address this challenge, we propose a decision support approach, called DESMILS in article [PIV], that we discuss in Chapter 5.

DESMILS allows the DM to provide his/her preferences to solve a single-item multiobjective lot sizing problem for a small number of selected items that have different preferences in lot sizing decisions. These items are carefully selected to represent a large number of items. The preferences obtained from the DM for the selected items are then accommodated in deriving lot sizes for the other items that have similar preferences. Therefore, the need of repeating a decision making process for each item separately is avoided, but we can accommodate different preferences for different items. DESMILS enables applying interactive multiobjective optimization methods in solving multi-item lot sizing problems. It can also be applied to any variant of single-item multiobjective lot sizing models. In the case study of Chapter 5, we demonstrate that the DM found lot sizes for 94 items with only 10 decision making processes.

The DM appreciated many benefits that he gained from the studies in this thesis. The single-item lot sizing models proposed in articles [PI]-[PIII], allow him to consider different important indicators simultaneously as objective functions in various lot sizing problems to measure the success of his day-to-day operations. Additionally, he found the usefulness of the safety order time concept proposed in article [PI] for his inventory control because it responds faster than safety stock, especially when demand fluctuates rapidly. Moreover, he gained valuable insights to be used in negotiations with the supplier with the proposed model and concept in article [PIII]. Furthermore, he really appreciated DESMILS which saved his time and effort in solving his lot sizing problems for a large number of items, where he was able to obtain the solutions that best represent his preferences by spending an acceptable amount of his time.

The rest of the thesis is structured as follows. Chapter 2 establishes the background concepts of multiobjective optimization used in this thesis. Chapters 3 and 4 introduce articles [PI]-[PIII], where the proposed lot sizing models are discussed in Chapter 3 followed by the case studies in Chapter 4. Chapter 5 introduces article [PIV], which describes DESMILS and the related case study. Finally, we provide our conclusions and put forward some future research directions in Chapter 6, followed by the author's contributions in the included articles and the final thoughts.

2 SOME CONCEPTS OF MULTIOBJECTIVE OPTIMIZATION

In this chapter, the concepts and terminologies related to multiobjective optimization to be used in this thesis are provided. Subsequently, we present some interactive methods for solving multiobjective optimization problems.

2.1 Basic Concepts

Multiobjective optimization problems can be modeled mathematically as:

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

Here, $k \geq 2$ objective functions, $f_i : S \to \mathbb{R}$, $i = 1, \ldots, k$, are to be optimized simultaneously. For generality, we consider all the objective functions to be minimized. If some of the objective functions f_i are to be maximized, they can be transformed to minimize $-f_i$.

A decision variable vector $\boldsymbol{x}$ consist of the decision variables $(x_1, \ldots, x_n)^T$. These vectors are feasible if they belong to 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 equality and inequality constraints and/or lower and upper bounds for the decision variable vectors. For a feasible decision variable vector $\boldsymbol{x} \in S$, the corresponding vector $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$ is called a feasible objective vector, which consists of objective function values calculated at $\boldsymbol{x}$. The feasible objective vector belongs to the feasible objective region $Z = \boldsymbol{f}(S)$, which is a subset of the objective space $\mathbb{R}^k$. In what follows, we use the term 'solution' to refer to an objective vector.

The objective functions f_i, $i = 1, \ldots, k$ are (at least partly) conflicting with each other. Therefore, there is no single optimal solution of multiobjective optimization problems (1) where all objective functions achieve their individual optima. To describe solutions of (1), we define the concept of dominance and nondominance as follows.

Definition 1 (Dominance and nondominance)**.** *Let $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ be two feasible solutions. We say that $\boldsymbol{z}^1$ dominates $\boldsymbol{z}^2$ if $z_i^1 \leq z_i^2$ for all $i = 1, \ldots, k$ and $z_j^1 < z_j^2$ for at least one index j. The solutions that do not dominate each other are called nondominated solutions.*

The solutions of (1) are a set of so-called Pareto optimal solutions defined as follows:

Definition 2 (Pareto optimality)**.** *A feasible solution $\boldsymbol{z}' \in Z$ and its corresponding decision variable vector $\boldsymbol{x}' \in S$ are Pareto optimal if no other feasible solution $\boldsymbol{z} \in Z$ dominates $\boldsymbol{z}^{'}$.*

Note that nondominated solutions are not always Pareto optimal, but Pareto optimal solutions must be nondominated. The set of all Pareto optimal solutions is called a Pareto optimal front. To describe the ranges of objective function values in the Pareto optimal front, we define its lower and upper bounds, which is called an ideal point $\boldsymbol{z}^*$ and a nadir point $\boldsymbol{z}^{nad}$, as follows.

Definition 3 (Ideal point)**.** *An ideal point represents the best values that objective functions can achieve in the Pareto optimal front. The components of the ideal point can be determined by minimizing each objective function individually, that is $z_i^* = \underset{\boldsymbol{x} \in S}{\text{minimize}}\ f_i(\boldsymbol{x}), i = 1, \ldots, k$*

Definition 4 (Nadir point)**.** *A nadir point represents the worst values of objective functions in the Pareto optimal front.*

In practice, the nadir point is difficult to calculate when the Pareto optimal front is unknown, and no reliable procedure is available to calculate it for problems with more than two objective functions [36]. Therefore, it is commonly approximated (see e.g. [10, 18, 32]). For computational reasons, we also define a utopian point $\boldsymbol{z}^{**}$, which is strictly better than the ideal point. The components of the utopian point are formed by $z_i^{**} = z_i^* - \epsilon, i = 1, \ldots, k$, where ϵ is a relatively small positive scalar.

All Pareto optimal solutions are equally acceptable mathematically. Therefore, additional information from a decision maker (DM) is needed to select one Pareto optimal solution as the final solution to be used in practice. A DM is an expert in the problem domain who has responsibility for making a decision in the application area; for example, in lot sizing, he/she is a supply chain manager in a company. The DM is responsible for providing preference information by relying on his/her deep knowledge of the problem.

Solving a multiobjective optimization problem means supporting the DM to find his/her most preferred solution among the Pareto optimal solutions. The process of finding the solution by the DM is called a decision making process. Besides the DM, another important role in solving a multiobjective optimization problem is an analyst, who is responsible for helping the DM to find the best solution based on his/her preferences. The analyst should have knowledge regarding the multiobjective optimization methods and has responsibility for the mathematical aspects of the model and making preparations for the decision making process.

Numerous methods have been proposed in the literature to solve multiobjective optimization problems (see e.g. [36, 37, 42]). In [36], these methods are divided into four classes according to the role of the DM in the decision making process. The classes are:

- no-preference methods, where no preference information from the DM is used;
- a priori methods, where the DM is first asked to specify his/her preferences, and a Pareto optimal solution that satisfies the preferences as well as possible is then found;
- a posteriori methods, where a (representative) set of Pareto optimal solutions is first generated and presented to the DM, and then he/she is expected to choose the most preferred one among them; and
- interactive methods, where the DM provides his/her preferences iteratively during the decision making process.

The DM can express his/her preferences in different ways [36]. One of them is using a reference point, which is defined as follows.

Definition 5 (Reference point)**.** *A reference point* $\bar{z} \in \mathbb{R}^k$ *is a vector consisting of aspiration levels. Aspiration levels* $\bar{z}_i$*,* $i = 1, \ldots, k$*, are the desirable values for each objective function given by the DM.*

Other ways to elicit preference information from the DM, for example, are using classification (e.g., which objective(s) to be improved and which one(s) to be impaired) or selecting one preferred solution from a set of Pareto optimal solutions.

2.2 Scalarizing Function

Scalarizing functions $s : \mathbb{R}^n \rightarrow \mathbb{R}$ are often used in many multiobjective optimization methods to transform multiobjective optimization problems into single objective problems:

$$\begin{aligned} &\text{minimize} \quad s(\boldsymbol{f}(\boldsymbol{x})) \\ &\text{subject to} \quad \boldsymbol{x} \in S. \end{aligned} \tag{2}$$

Any appropriate single objective optimization method can be used to optimize (2). The optimal solution of (2) is a solution of problem (1), and some scalarizing functions guarantee the Pareto optimality of the solution obtained [62]. Scalarizing functions typically include the preference information from the DM, for example, by using the reference point.

Different scalarizing functions have been suggested in the literature to be used in different multiobjective optimization methods [39]. The achievement scalarizing function (ASF), which is introduced in [60], is a widely used scalarizing function. The basic principle of ASF is to find the closest Pareto optimal solution to the reference point. This function works well to find a Pareto optimal solution for the multiobjective optimization problem (1) for any reference point $\tilde{z} \in \mathbb{R}^k$, regardless of whether it is in the feasible objective region or not.

Different variants of ASFs have been introduced in the literature [39, 45]. One of them, which is used in this thesis, can be written as follows:

$$s(\boldsymbol{f}(\boldsymbol{x})) = \max_{i=1,\dots,k} \left\{ \frac{f_i(\boldsymbol{x}) - \tilde{z}_i}{z_i^{nad} - z_i^{**}} \right\} + \rho \sum_{i=1}^{k} \frac{f_i(\boldsymbol{x})}{z_i^{nad} - z_i^{**}}, \tag{3}$$

where $\rho > 0$ is a relatively small positive scalar. By solving problem (2) with the ASF (3), a Pareto optimal solution for problem (1) can be obtained, and different Pareto optimal solutions can be found by changing $\tilde{z}$ [36, 37].

2.3 Interactive Methods

Among the four classes of multiobjective optimization methods, interactive methods have been found useful in supporting the DM to find his/her most preferred solution [38]. The benefits of these methods have been discussed in several studies (see e.g. [1, 38, 42, 61]). These methods provide possibilities for the DM to learn about the problem and the relationship among the objective functions. During the decision making process, the DM is allowed to explore different solutions and change his/her preferences until he/she finds the best solution for him/her. In this way, the DM should be more aware of the nature of the problem before deciding on one Pareto optimal solution as the final solution. These methods are efficient from both computational and cognitive points of view, because the DM can concentrate on the solutions that are interesting for him/her and only these interesting solutions need to be generated.

In general, interactive methods start by presenting some information to the DM, such as ideal and nadir points. Then, the DM is asked to provide some preference information (e.g., a reference point). One or more (Pareto optimal) solution(s) are then generated corresponding to the DM's preferences and shown to the DM. If several solutions are generated, the method asks the DM to select the best solution so far. The DM is then iteratively asked to provide his/her preferences, for example, by providing a reference point or selecting one solution among the generated Pareto optimal solutions, depending on the method in question, and new Pareto optimal solution(s) are generated according to the DM's preferences and presented to the DM. These iterative processes are repeated until the DM has found the most preferred solution.

Many interactive methods have been proposed in the literature using different ways in eliciting preference information from the DM and various techniques in generating Pareto optimal solutions [36, 38]. In what follows, we explain three of them which are used in this thesis. They are the E-NAUTILUS method [50], the NAUTILUS Navigator method [49], and the synchronous NIMBUS method [40]. Graphical user interfaces for these methods are available in the DESDEO framework [43], which is an open-source software implemented in Python.

2.3.1 E-NAUTILUS

The enhanced NAUTILUS (E-NAUTILUS) method is introduced in [50]. This is a variant of NAUTILUS methods [41], which were designed to avoid trading-off among the objective functions. According to the prospect theory [30], people do not respond equally to gains and losses; instead, they tend to fear losses more than desire gains. Therefore, the DM may fail to find his/her desirable solution because moving from one Pareto optimal solution to another one may cause some decisional stress to the DM [33]. Following this philosophy, NAUTILUS methods do not start from the Pareto optimal solution(s) as most other interactive methods do, but from the nadir point as the worst objective function values. Then, in each iteration, all objective function values are improved from the previous iteration until the preferred Pareto optimal solution is found as the final solution. In this way, the DM can have a free search without requiring any trade-offs and, iteratively, gain in all objective functions.

The E-NAUTILUS method is suitable for computationally expensive problems. This method has the three following stages:

1. pre-processing stage, where a set of (Pareto optimal) solutions is generated;
2. interactive decision making stage, where the DM gives his/her preferences iteratively until he/she obtains a preferred solution (among the pre-generated set) in the last iteration; and
3. post-processing stage, which is needed to ensure the Pareto optimality of the final solution, when the method used in the first stage cannot guarantee it.

The pre-processing stage is the most time-consuming part of E-NAUTILUS, especially for computationally expensive problems. However, it is done without the involvement of the DM. Any a posteriori methods can be used in this stage to generate a set of Pareto optimal solutions or a set of nondominated solutions that approximates Pareto optimal solutions. We denote this set as $P(0)$. Here, an analyst should have knowledge of an appropriate (a posteriori type) method to generate a sufficient number of solutions. Then, the (estimated) ideal and the nadir points are calculated in this stage by finding the best and the worst values of each objective function from the solutions in $P(0)$.

The DM is only involved in the second stage, called the interactive decision making stage. This stage uses the generated solutions $P(0)$ without solving the original multiobjective optimization problem that can be computationally expensive. Therefore, there are no time-consuming computations involved, and the decision making process can be conducted without waiting times. At the beginning of this stage, the estimated ideal and nadir points ($\boldsymbol{z}^*$ and $\boldsymbol{z}^{nad}$) are presented to the DM, and he/she is asked to provide the number of iterations N_I and the number of candidates N_S that he/she wants to compare at each iteration. We refer to a candidate as a vector in the objective space $\mathbb{R}^k$, which does not necessarily correspond to any decision variable vector.

The decision making process is conducted in N_I iterations, and three kinds of information are provided to the DM in each iteration. They are: N_S candidates, the best reachable values of each candidate, and the closeness of the candidates to the Pareto optimal front. We define the concept of a reachable solution as follows.

Definition 6 (Reachable solution)**.** *Let $\mathbf{z'} \in \mathbb{R}^k$ be a candidate. A feasible solution $\mathbf{z} \in Z$ and the corresponding decision variable vector $\mathbf{x} \in S$, are reachable from $\mathbf{z'}$ if $\mathbf{z}$ dominates $\mathbf{z'}$.*

The best reachable values of candidate $\mathbf{z'}$ are the best values of each objective function that can be achieved by reachable solutions of $\mathbf{z'}$. The DM is then asked to select one preferred candidate in each iteration ($\mathbf{z}(h)$), after comparing N_S candidates based on information provided to him/her. The set of reachable solutions ($P(h)$) is then updated by deleting solutions that are not reachable from ($\mathbf{z}(h)$).

The candidates of iteration h, $\mathbf{z}(h,i)$, $i = 1,\ldots,N_S$ are calculated based on the previous preferred candidate $\mathbf{z}(h-1)$, with the starting point $\mathbf{z}(0) = \mathbf{z}^{nad}$. Each candidate represents different directions to move toward the Pareto optimal front, and selecting one preferred candidate directs him/her to the direction that he/she likes. For each candidate, the worst objective function values of its reachable solutions is the candidate itself, and the best reachable values are calculated by solving the following ε-constraint problem [24] for $r = 1,\ldots,k$:

$$\begin{aligned} \text{minimize} \quad & f_r(\mathbf{x}) \\ \text{subject to} \quad & f_j(\mathbf{x}) \leq z_j(h,i),\ j = 1,\ldots,k, j \neq r \\ & \mathbf{x} \in P(h). \end{aligned} \tag{4}$$

The ranges between the best and worst objective function values are called reachable ranges.

During the iterations, the candidates get closer to the Pareto optimal front, and the reachable ranges become smaller. The closeness of each candidate to the Pareto optimal front is shown to the DM as a percentage and calculated as follows:

$$d(h,i) = \frac{\|\mathbf{z}(h,i) - \mathbf{z}^{nad}\|}{\|\bar{\mathbf{z}}(h,i) - \mathbf{z}^{nad}\|} \times 100\%,\ i = 1,\ldots,N_S. \tag{5}$$

The result of the interactive decision making stage is $\mathbf{z}(N_I) \in P(0)$, which is a nondominated solution that best represents the DM's preferences.

The Pareto optimality of $\mathbf{z}(N_I)$ depends on the a posteriori method used in the first stage. However, some methods, such as evolutionary methods, cannot guarantee that their solutions are Pareto optimal. Therefore, the last stage is needed in this case, to ensure the Pareto optimality of the final solution. To get a final Pareto optimal solution, $\mathbf{z}(N_I)$ is projected onto the Pareto optimal front by minimizing ASF (3) with $\mathbf{z}(N_I)$ as a reference point. A more detailed algorithm of the E-NAUTILUS method can be seen in [50], and an explanation of the interface for E-NAUTILUS can be found in [PI].

2.3.2 NAUTILUS Navigator

The NAUTILUS Navigator method [49] is another variant of NAUTILUS methods [38]. All methods in the NAUTILUS family enable the DM to have a free search without trading-off by starting from the worst objective function values, but they differ in the way used to interact with the DM to find the final solution and how solutions are generated in each iteration. NAUTILUS Navigator uses navigation ideas elaborated in [26] to direct the movement from the worst objective function values as the starting point to a Pareto optimal solution as the final solution.

Similar to the E-NAUTILUS method, NAUTILUS Navigator needs a set of Pareto optimal solutions or a set of nondominated solutions that approximates Pareto optimal solutions to be generated before the navigation process starts. This makes NAUTILUS Navigator suitable for computationally expensive problems, because the process of generating solutions, which may take time because of expensive functions, is done without involving the DM. Therefore, the DM is allowed to navigate in real time without waiting times. The concept of the reachable range is also used in this method, where, during the navigation process, the DM can see the ranges between the best and the worst objective function values of the reachable solutions.

At the beginning of the navigation process, the estimated ideal and nadir points, which are derived from the pre-generated set of solutions, are presented to the DM, to see the reachable ranges of each objective function in the first step. The DM specifies his/her preferences by providing a reference point as a search direction to direct the movement towards desired Pareto optimal solutions. Each component of the reference point provided by the DM (aspiration level), must lie within the reachable ranges.

The navigation process continue from step to step unless the DM stops it, and there are 100 steps by default until a Pareto optimal solution is reached. The DM can change the speed of movement from 1 (minimum speed) to 5 (maximum speed), which means the number of steps taken per second by the algorithm. During the navigation process, the connection to the decision space is temporarily loose, but at the end, a Pareto optimal solution and the corresponding decision variable vector in the decision space are presented to the DM.

NAUTILUS Navigator moves towards the Pareto optimal front from step to step in the direction specified by the DM and at the speed he/she defines. The DM can see the changes in the reachable ranges while they shrink from step to step in the direction of the reference point. Whenever he/she wants to change direction, he/she can stop at any step and go back to any previous step to provide a new reference point. The DM also can specify bounds for the objective functions that must not be exceeded. The DM can navigate until he/she finds his/her most preferred Pareto optimal solution at the end of the navigation process, where the reachable ranges shrink to a single point.

Similar to E-NAUTILUS, the Pareto optimality of the solution obtained depends on the method used to generate the set of solutions before the navigation process. Therefore, the last stage of E-NAUTILUS, where we project the solution onto the Pareto optimal front, can be applied here if needed. The detailed algorithm of the NAUTILUS Navigator method can be seen in [49].

2.3.3 Synchronous NIMBUS method

Different from NAUTILUS methods, the synchronous NIMBUS method [40] is a trade-off based method where the DM needs to deal with trade-offs in each iteration to move from a Pareto optimal solution to another one to find the most preferred Pareto optimal solution. In this method, the DM gives her/his preferences by using a classification, which is a natural way of expressing preference information [35], without any artificial concepts needed. Many studies have applied this method to support the DM in solving various real-world problems (see e.g. [51, 52, 55]).

Before starting the interactive process, NIMBUS needs a Pareto optimal solution to be provided, e.g. by the DM, as a starting point. Otherwise, a Pareto optimal solution located approximately in the middle of the Pareto optimal front can be used as a starting point. This point, called a neutral compromise solution, can be calculated by solving the ASF (3) with aspiration levels $\tilde{z}_i = (z_i^{nad} + z_i^{**})/2$, $i = 1, \ldots, k$.

In the first iteration, the starting point is presented to the DM together with the ideal and nadir points. The DM then needs to indicate what kind of changes in the objective function values of the current solution are needed to obtain a more preferred Pareto optimal solution. Therefore, in each iteration, he/she is asked to classify each of the objective functions into one of the five following classes:

1. $I^{<}$: if he/she wants to improve the current value,
2. $I^{\leq}$: if he/she wants to improve the current value to a certain level,
3. $I^{=}$: if he/she wants to keep the current value,
4. $I^{\geq}$: if he/she allows to impair the current value until a certain bound, and
5. $I^{\diamond}$: if he/she lets the current value change freely.

To move from a Pareto optimal solution to another, at least one objective function must be impaired to get better value(s) of other objective function(s). Therefore, a classification is feasible only if $I^{<} \cup I^{\leq} \neq \emptyset$ and $I^{\geq} \cup I^{\diamond} \neq \emptyset$, and all objective functions have been classified ($I^{<} \cup I^{\leq} \cup I^{=} \cup I^{\geq} \cup I^{\diamond} = \{1, \ldots, k\}$). If classes $I^{\leq}$ and/or $I^{\geq}$ are selected, the bounds must be specified.

After having provided the classification information, the DM can specify the maximum number of solutions (one to four) that he/she wants to consider and compare. Then, the desired number of new Pareto optimal solutions, that reflect the DM's preferences as well as possible, are generated by using different scalarizing functions. The new Pareto optimal solutions are then presented to the DM, and he/she selects one preferred solution among them. If he/she is satisfied with this solution, he/she can stop with it as the final solution. Otherwise, he/she can continue to the next iteration and use this solution as the starting point of a new classification. The DM may also request a desired number of intermediate solutions to be generated between any two interesting solutions obtained so far. Further details about the Synchronous NIMBUS method can be seen in [40].

3 SINGLE-ITEM MULTIOBJECTIVE LOT SIZING MODELS

To bridge the gap between theory and the challenges of real industrial problems, in this thesis, we solve lot sizing problems that are integrated with different problems. These problems are motivated by real challenges in a manufacturing company. In this chapter, we first define the general assumptions and notations which are used throughout this thesis. We then present several lot sizing models that we proposed in the different articles constituting this thesis ([PI]-[PIII]).

3.1 General assumptions and notations

We consider a single-item multi-period lot sizing problem, which means that more than one period is considered, and an order should be placed in each period. The replenishment process follows a periodic review policy, where orders are reviewed over discrete time periods $t = 1, \ldots, T$. In each period t, the order quantity $Q(t)$ is reviewed at the beginning of the period, and the order arrives after a certain lead time L. The lead time is assumed to be stochastic for the model in Section 3.3 and constant for the others.

We make the following assumptions in general, while the specific assumptions are explained in each subsection if needed.

1. All the input data are assumed to be ready to use.

2. The predicted demand during period t, denoted by $D(t)$, is assumed to be stochastic, independent of other periods, and follows a normal distribution with a mean μ and a standard deviation σ.

3. The cost to place one order is c, without any capacity limit.

4. No backorder cost is considered in the models.

5. There is a requirement from the supplier to place an order with a minimum order quantity moq, and it must be rounded up by a rounding value r because of the packaging size. Therefore, we can only place an order by following the formula $moq + a\,r$, for any integer $a \geq 0$.

The notations used in this paper are defined as follows (note that, here, moq is a decision variable representing the amount of MOQ).

t	index of time period, $t = 1, \ldots, T$
p	price to purchase one unit of item
c	cost to place one order
h	cost to hold one unit of item for one period
L	lead time
$D(t)$	predicted demand during period t
μ	average demand
σ	standard deviation of demand for one period
moq	minimum order quantity
r	rounding value
$Q(t)$	lot size or order quantity at period t
$Y(t)$	order indicator
$I(t)$	inventory position at the end of period t

The value $Y(t)$ follows the formula:

$$Y(t) = \begin{cases} 0, & \text{if } Q(t) = 0 \\ 1, & \text{otherwise,} \end{cases}$$

while the value of $I(t)$ follows the formula:

$$I(t) = I(t-1) + Q(t - \lfloor L \rfloor) - D(t),$$

where $\lfloor L \rfloor$ is the biggest integer lower than or equal to L.

3.2 Lot sizing with safety strategy placement

In article [PI], we study lot sizing in a stochastic environment on demand. Many companies usually use a safety stock (SS) to protect against demand uncertainty (see e.g. [22, 23, 44]). A SS is defined as a level of the item that is kept in their inventory as a buffer to avoid stockout when demand exceeds the forecast. However, a static SS that keeps a fixed level of stock regardless of demand fluctuation can lead to stockout during periods of high demand [9]. Some researchers used a dynamic SS that can be changed dynamically from period to period to handle high fluctuations in demand [28, 46]. However, a dynamic SS is not suitable for lot sizing problems with large numbers of decision variables and various types of practical production constraints [57].

We propose a so-called safety order time (SOT) as another way to handle demand uncertainty. With a SOT, the stock that is kept to handle demand uncertainty is calculated based on time. For example, one week's worth of demand is always kept in the inventory when we set SOT as one week. Because the demands fluctuate, the stock is also changed dynamically. Therefore, we can better manage high peaks of demand with SOT. In this way, a SOT fills the need of having dynamic stock in an easy and efficient way so that it can be applied to lot sizing problems with large numbers of decision variables and constraints. In the model proposed in article [PI], we use both SS and SOT as a safety strategy to increase the preparedness of handling demand uncertainty.

We formulate a multiobjective optimization model with four objective functions to solve the defined lot sizing problem. The objective functions are: minimize purchasing and ordering cost (POC), minimize holding cost (HC), maximize cycle service level (CSL), and maximize inventory turnover (ITO). The proposed model [PI] can be written as:

$$
\begin{aligned}
\text{minimize} \quad & POC = \sum_t Q(t)\, p + \sum_t Y(t)\, c, \\
& HC = \sum_t \frac{I(t-1) + I(t)}{2}\, h, \\
\text{maximize} \quad & CSL = F\left(\frac{SS + \mu\, SOT}{\sigma}\right), \\
& ITO = \sum_t \frac{D(t) + \sigma}{(I(t-1) + I(t))/2}, \\
\text{subject to} \quad & FR = \frac{I(t-1) + \sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{\sum_{j=t}^{t+\lfloor P \rfloor} D(j) + (P - \lfloor P \rfloor) D(\lceil P \rceil)} \geq 1 \text{ , for } t = 1, \ldots, T, \\
& Q(t) = Y(t)\,(moq + a\, r) \text{ , for any integer } a \geq 0 \text{ and } t = 1, \ldots, T, \\
& I(t) \geq SS + SOT\, D(t) \text{ , for } t = 1, \ldots, T, \\
& SS \geq 0 \text{ and } SOT \geq 0,
\end{aligned}
\tag{6}
$$

where $P = L + SOT$. By solving model (6), the DM can determine the optimal order quantity for each period $Q(t), t = 1, \ldots, T$, simultaneously with SS and SOT, with his/her most preferred solution that best balances among POC, HC, CSL, and ITO.

The cost functions are adapted from the dynamic economic order quantity model [58], but we separate POC and HC as the first and the second objective functions because they show different behavior in the inventory system [47]. Therefore, there is a clear trade-off between POC and HC. For instance, when considering the same total order quantity over the period considered, ordering more frequently results in higher POC but lower HC compared to ordering less frequently. Furthermore, POC and HC typically serve different purposes, where POC focuses on procurement efficiency and HC is related to warehousing management. Considering them as separate objective functions enables the DM to align his/her strategies with specific goals and priorities.

The CSL is set to be the third objective function to evaluate the effectiveness of the proposed safety strategy in dealing with unexpected demand within one period. A CSL is defined as the probability of not having a stockout in a replenishment cycle [15]. We modify the CSL formula defined in [56] to cover both SS and SOT. In this case, to cover unpredicted demand for one period, we have additional stock in the amount of SS and the average demand for SOT time units. The CSL has trade-offs with POC and HC. Maximizing CSL requires purchasing additional stock, which naturally increase both POC and HC.

The ITO is set to be the last objective function because it is an important indicator of lot sizing in practice. It reflects the amount of an item sold or used in production over a year, which provides insight into how quickly the company is selling or using the item. With ITO, the DM can evaluate the effectiveness of his/her inventory management strategies. ITO can be calculated as the ratio of item usage to the average inventory level. In model (6), we modify the traditional ITO formula [54] to be suitable for our problem.

From model (6), one can see a connection that maximizing ITO implies minimizing HC. Thus, it is possible to consider only one of them, if it is preferred by the DM (since the problem formulation must have objective functions that the DM in question has interest in). In this thesis, we consider both HC and ITO as objective functions because they are both important indicators that can provide different meanings for the DM. ITO is an indicator of the efficiency of the inventory management system, while HC is an indicator of costs associated with warehousing management. They both are needed when the DM as a supply chain manager reports to the top management. By examining both indicators, the DM can make informed decisions that strike a balance between operational and warehousing efficiency.

3.3 Lot sizing with safety stock and safety lead time

In article [PII], we consider both demand and lead time uncertainty in a lot sizing problem. Therefore, in this article, the lead time is assumed to be stochastic and follows a normal distribution with a mean L and a standard deviation s. As mentioned earlier, a SS is widely used to handle demand uncertainty, and to handle lead time uncertainty, an additional time period, called a safety lead time (SLT), is commonly used [59]. With SLT, companies keep their stocks available to satisfy the demand during the SLT period so that they can avoid stockout when the order is delayed.

The problem of determining an optimal value for SS has been widely studied [20]. Various methods have been proposed to find an optimal SS that ensures a high service level while satisfying demand and minimizing costs [53]. However, finding an optimal value for SLT has not been studied as extensively, and, to the best of our knowledge, there is no formula available for measuring the quality of SLT.

We propose a formula called the probability of product availability (PPA) to evaluate the quality of SLT. This formula calculates the probability of not having stockout due to the late delivery. As said, orders are expected to arrive after a stochastic lead time period L, and we ensure stock availability in this period as well as an additional SLT period. Therefore, by maximizing the PPA formula we aim to increase the probability of the actual order arrives during the period $L + SLT$. The proposed PPA formula can be written as follows:

$$\begin{aligned} PPA &= P(\text{actual delivery time} \leq L + SLT) \\ &= F(L + SLT, L, s) = F\left(\frac{SLT}{s}\right). \end{aligned} \tag{7}$$

We formulate a multiobjective optimization model with six objective functions to solve the integration problem of lot sizing and determination of SS and SLT. The objective functions are: minimize purchasing cost (PC), ordering cost (OC), and holding cost (HC), as well as maximize CSL, PPA, and ITO. The multiobjective optimization model proposed in [PII] can be written as:

$$\begin{aligned}
\text{minimize} \quad & PC = \sum_t Q(t)\ p, \\
& OC = \sum_t Y(t)\ c, \\
& HC = \sum_t \frac{I(t-1) + I(t)}{2}\ h, \\
\text{maximize} \quad & CSL = F\left(\frac{SS}{\sqrt{\sigma^2(1+SLT) + \mu^2 s^2}}\right), \\
& PPA = F\left(\frac{SLT}{s}\right), \\
& ITO = \sum_t \frac{D(t)}{(I(t-1) + I(t))/2}, \\
\text{subject to} \quad & FR = \frac{I(t-1) + \sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{\sum_{j=t}^{t+\lfloor P \rfloor} D(j) + (P - \lfloor P \rfloor) D(\lceil P \rceil)} \geq 1\ , \text{for } t = 1, \ldots, T, \\
& Q(t) = Y(t)\ (moq + a\, r)\ , \text{for any integer } a \geq 0 \text{ and } t = 1, \ldots, T, \\
& I(t) \geq SS + SLT\, D(t)\ , \text{for } t = 1, \ldots, T, \\
& SS \geq 0 \text{ and } SLT \geq 0,
\end{aligned} \tag{8}$$

where $P = L + SLT$.

Similar to model (6), in model (8), we adapt the cost functions from the dynamic lot sizing problem [58]. However, we consider all of them as separate objective functions here to see the trade-offs more clearly. We then consider CSL to be maximized to prevent stockout due to demand uncertainty, and, to avoid stockout due to late delivery, the PPA formula is considered to be maximized. Furthermore, we maximize ITO in the last objective function to measure the effectiveness of the inventory system. By solving model (8), the DM can determine the optimal order quantity for each period ($Q(t), t = 1, \ldots, T$) simultaneously with SS and SLT as the decision variables.

3.4 Lot sizing with minimum order quantity determination

Article [PIII] focuses on the integration of lot sizing and minimum order quantity (MOQ). A MOQ is often imposed in practice by suppliers on their products to ensure their production and ordering process. This requirement is commonly set as a constraint, such as in articles [PI] and [PII]. However, to the best of our knowledge, there has been no study in the literature focused on the problem of determining a MOQ from the buyer's perspective. Therefore, to support the DM as a buyer in negotiating a preferred MOQ with the supplier, besides order quantities, we also determine MOQ as a decision variable in the model that we proposed in article [PIII].

As compensation of a MOQ, suppliers commonly offer quantity discounts to pursue buyers to order in large quantities. There are many types of discounts that have been considered in the literature (see e.g. [5, 63]). One of them is an all-units discount that we use in this thesis. This type of discount reduces prices for every unit purchased if the order quantity exceeds a threshold. In this thesis, we use three prices (p_1, p_2, p_3) for different order quantities, where a higher order quantity, a lower price is applied to the order. The price function [63] can be written as follows:

$$P(t) = \begin{cases} p_1 & \text{if } Q(t) < a_1 \\ p_2 & \text{if } a_1 \leq Q(t) < a_2 \\ p_3 & \text{if } Q(t) \geq a_2, \end{cases} \tag{9}$$

where $p_1 > p_2 > p_3$ and $a_1 < a_2$.

To measure the quality of MOQ in satisfying demand, we propose the MOQ level formula, which provides a ratio between the MOQ and the expected demand, as follows:

$$ML = \frac{moq}{\mu}.$$

Then, to solve the defined lot sizing problem, we formulate a multiobjective optimization model [PIII] with five objective functions, that can be written as follows.

$$
\begin{aligned}
\text{minimize} \quad & ML = \frac{moq}{\mu}, \\
& PC = \sum_t Q(t)\ P(t), \\
& OC = \sum_t Y(t)\ c, \\
& HC = \sum_t \frac{I(t-1)+I(t)}{2}\ H(t), \\
\text{maximize} \quad & ITO = \sum_t \frac{D(t)}{(I(t-1)+I(t))/2}, \\
\text{subject to} \quad & FR = \frac{I(t-1)+\sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{\sum_{j=t}^{t+\lfloor L \rfloor} D(j) + (L-\lfloor L \rfloor) D(\lceil L \rceil)} \geq 1 \text{ , for } t = 1,\ldots,T, \\
& Q(t) = Y(t)\ (moq + a\, r) \text{ , for any integer } a \geq 0 \text{ and } t = 1,\ldots,T, \\
& SS \geq 0 \text{ and } moq \geq 0.
\end{aligned}
\tag{10}
$$

In this model, the MOQ level formula is set as the first objective function. Then, for the same reason as model (8), we set PC, OC, and HC in separate objective functions and consider ITO as one of the objective functions. The main goal of this model is to find the order quantity for each period $Q(t), t = 1, \ldots, t_n$, together with the optimal MOQ values (*moq*) with the best balance among the objective functions.

The cost for holding one unit of the item is usually calculated as a percentage of the price (see e.g. [25]). Because of different prices for different order quantities, here, we calculate this cost proportionally by using the following formula:

$$
H(t) = iir * \frac{\sum_t Q(t)\ P(t)}{\sum_t Q(t)}, \tag{11}
$$

where *iir* is the internal interest rate for one period.

In the case study of article [PIII], we combine lot sizing models (6) and (10) to be able to handle lot sizing simultaneously with MOQ, SS, and SOT. The multiobjective optimization model can be written as follows.

$$
\begin{aligned}
\text{minimize} \quad & ML = \frac{moq}{\mu}, \\
& PC = \sum_t Q(t)\ P(t), \\
& OC = \sum_t Y(t)\ c, \\
& HC = \sum_t \frac{I(t-1) + I(t)}{2}\ H(t), \\
\text{maximize} \quad & CSL = F\left(\frac{SS + \mu\, SOT}{\sigma}\right), \\
& ITO = \sum_t \frac{D(t)}{(I(t-1) + I(t))/2}, \\
\text{subject to} \quad & FR = \frac{I(t-1) + \sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{\sum_{j=t}^{t+\lfloor P \rfloor} D(j) + (P - \lfloor P \rfloor) D(\lceil P \rceil)} \geq 1 \text{ , for } t = 1, \ldots, T, \\
& Q(t) = Y(t)\ (moq + a\, r) \text{ , for any integer } a \geq 0 \text{ and } t = 1, \ldots, T, \\
& I(t) \geq SS + SOT\, D(t) \text{ , for } t = 1, \ldots, T, \\
& SS \geq 0, SOT \geq 0, \text{ and } moq \geq 0,
\end{aligned}
\tag{12}
$$

where $P = L + SOT$.

4 REAL CASE STUDIES FOR SOLVING LOT SIZING PROBLEMS

After introducing different lot sizing models, in this chapter, we demonstrate their applicability in solving three real case studies from a manufacturing company. The company in question is a semi-heavy vehicle company which primarily relies on high volume assembly line technology for its production. Material management plays a crucial role within this company's operations. The company provided real data from its ERP system as input data. The supply chain manager of the company acted as the DM in these case studies to provide his preferences into the decision making processes based on his expertise, and ensure the validity of the results.

4.1 Description of case studies

We investigate three different items, where one item is applied for one model in Chapter 3. They are called item 1, item 2, and item 3, which are used in case studies 1, 2, and 3, respectively, in this chapter. As additional information, item 1 is a pneumatic component, item 2 is a mechanical transmission component, and item 3 is a fastener utilized for electric components. We visualize the demand data for all items in Figure 1, while other input data are presented in Table 1. As can be seen in this table, item 3 has different prices as compensated of a MOQ in model (12) which is considered in case study 3.

All the items are reviewed in a weekly planning horizon. We consider 48 weeks for item 1, 41 weeks for item 2, and 24 weeks for item 3. The different lengths of the periods are set based on the request from the DM for specific case studies. During the lead time period, the company has made previous orders ($Q(t), t = 1, .., L$): [420, 70, 140, 210, 140, 140] for item 1, [48, 119, 120, 120, 48, 96] for item 2, and [0, 0] for item 3. Here, the different lengths are caused by the different lead time of each item.

The cost to hold one unit of the item for one period is ten percent of its price

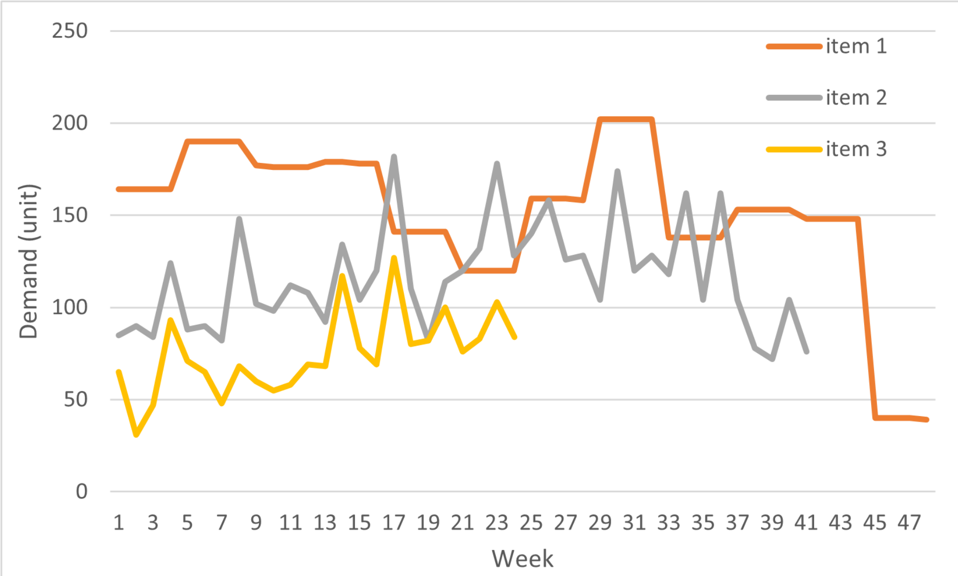


FIGURE 1 Demand data for each item

TABLE 1 Input data for the case studies

	item 1	**item 2**	**item 3**
price per unit p	€134	€91.18	€34.815 for $Q(t) < 360$ €27.85 for $360 \leq Q(t) < 500$ €23.21 for $Q(t) \geq 500$
cost to place one order c	€200	€200	€200
minimum order quantity moq	70 units	48 units	-
rounding value r	14 units	48 units	20 units
opening inventory $I(0)$	596 units	312 units	600 units
lead time L	6 weeks	6 weeks	2 weeks

annually. For item 3 in case study 3, we use formula (11) with an internal interest rate of 10% annually ($iir = 10\%/52$).

4.2 Generating a set of nondominated solutions

As said, lot sizing problems have been identified as computationally challenging problems [3, 11]. Therefore, in each case study, we generated a large number of nondominated solutions that approximate the Pareto optimal front to ensure that the decision making processes could be conducted without waiting time. Because of the complexity of lot sizing problems, many researchers have used evolutionary algorithms to solve various problems in this field [21, 29]. In the case studies discussed in this chapter, we used NSGA-III [17], which is an a posteriori multiobjective evolutionary algorithm to generate nondominated solutions. This method is suitable for multiobjective optimization problems with more than three objective functions. We applied an open-source framework called pymoo [12] because of its ability to consider integer variables and many constraints.

Generating a large number of nondominated solutions for the lot sizing problems defined in Chapter 3 is challenging. Because of several constraints and integer decision variables, the number of nondominated solutions generated by a single run of NSGA-III is limited. To obtain more solutions, we run this method several times using different sizes of the initial populations. We utilized the structured approach proposed in [16] to generate well-spaced initial populations. This approach uses the number of partitions to determine the number of points to be sampled, and we used the number of partitions from 1 to 20. We then combined the generated solutions by deleting the recurring and dominated solutions.

Because the number of generated solutions was still relatively low for case studies in Sections 4.4 and 4.5, we also rerun the NSGA-III method by varying parameters of evolutionary operators that were available in the pymoo framework. We applied different types of crossover operators for integer variables. We used a crossover probability of 0.9 for simulated binary crossover, uniform crossover, half uniform crossover, and four-point crossover, while, for the exponential crossover, we used a probability of 0.95. For mutation, we used polynomial mutation for integer variables with a mutation probability of 0.9. These parameters were selected after several experiments, and we found that they were suitable for our case studies. For other parameters, the default values in pymoo were used.

4.3 Case study 1: Determining lot sizes and safety strategy with E-NAUTILUS

In article [PI], we supported the DM to solve a lot sizing problem with safety strategy placement for item 1. We solved model (6) by using the E-NAUTILUS method. Before starting the decision making process, the DM wanted to limit the solutions by setting an additional constraint as follows:

$$SOT + \frac{SS}{\mu} \leq MS, \tag{13}$$

where MS refers to the maximum number of periods that the safety strategy can cover, and we set $MS = 1$ week for item 1.

As said, a set of nondominated solutions needs to be generated in the first stage of E-NAUTILUS. Here, we ran NSGA-III several times, as described in section 4.2, and obtained 651 nondominated solutions to be used in the interactive decision making stage. We estimated ideal and nadir points by recording best and worst values among these generated solutions and obtained z^* = (747 820, 2 717.24, 1.0, 252.96) and z^{nad} = (1 046 028, 9 133.52, 0.5, 13.66), respectively.

We presented the ideal and nadir points to the DM and asked him to provide the number of iterations to be conducted and the number of candidates to be considered in each iteration. Because the range of objective function values in the

TABLE 2 Summary of the decision making process with E-NAUTILUS for item 1

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

Pareto optimal front was wide, he preferred to conduct ten iterations, so that the candidates would not approach the Pareto front too fast. He did not want to miss some potentially interesting candidates during the decision making process. He wanted to consider four candidates in each iteration. In each iteration (h), the DM was provided with four candidates as desired and their best reachable values, which are presented in Table 2. The selected solution in each iteration is denoted in bold.

Table 2 indicates that in iterations one to three, the DM focused on ITO when he found the other objective function values acceptable. He selected the first candidate in the first iteration, the second candidate in the second iteration, and the third candidate in the third iteration to obtain the best values of ITO while he still had an opportunity to improve on the other objective functions in the next iterations. In iterations four to seven, the DM changed the direction and paid more attention to the CSL value because the current ITO values satisfied him. He preferred the second candidate in iteration 4 and the third candidates in iterations 5, 6, and 7 since all of them had the best CSL values.

In iterations eight and nine, the DM considered both ITO and CSL values. He decided to select the second candidates in these iterations because they had the best CSL and pretty good ITO values. Finally, in the last iteration, the DM was satisfied with CSL and ITO values. He decided to consider both the cost values in this iteration and selected the fourth candidate to get the best POC. He realized that this candidate had the worst HC value, but it was relatively close to the other candidates.

The last stage of E-NAUTILUS was conducted to ensure the Pareto optimality of the final solution because we used an evolutionary algorithm in the first stage. In this stage, we projected the nondominated solution selected from the previous stage $z(10)$ = (810 528, 3 355.80, 0.90, 54.48) onto the Pareto optimal front. For this purpose, we minimized (3) with $z(10)$ as a reference point. We solved this optimization problem by using a branch and bound method [34], which is commonly used for solving optimization problems with integer variables. We then obtained the final solution z_{final} = (753 848, 2 329.41, 0.924, 89.18), which improved from $z(10)$.

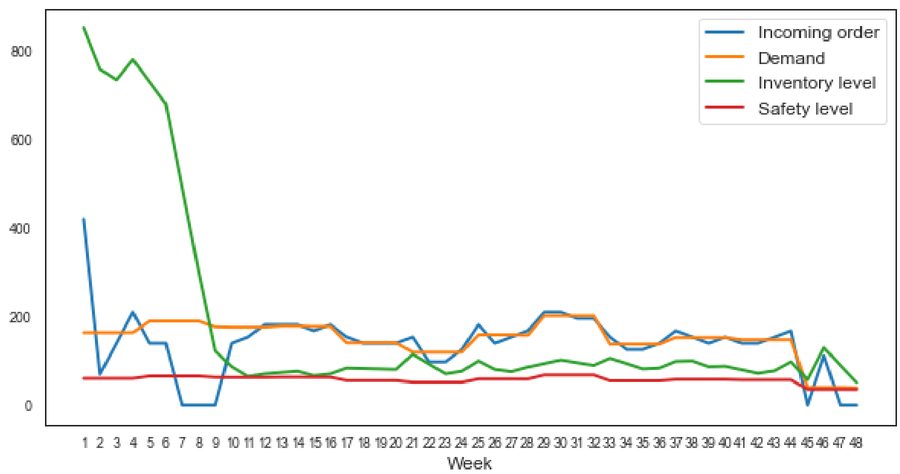


FIGURE 2 Optimized order quantities, demand, inventory level, and safety level for item 1 in the time period considered

The decision variable values corresponding to z_{final} were $SS = 28$ units, $SOT = 1$ day, and order quantities presented in Figure 2. In this figure, the blue line represents the incoming order quantities for each week, which is $Q(t - L)$ for $t = 7, .., 48$ and the previous order data for $t = 1, .., 6$. For comparison, we also present the demand data in the orange line. The inventory level and the safety level are provided in the green and the red lines, respectively, presenting that the inventory level is always larger than the safety level for every week. Therefore, with this final solution, the company had sufficient stock to handle demand uncertainty by at least the same amount as the safety level.

4.4 Case study 2: Determining lot sizes with safety stock and safety lead time with NAUTILUS Navigator

In this case study, which is described in article [PII], we considered a lot sizing problem with SS and SOT for item 2. We used NAUTILUS Navigator to support the DM in finding his most preferred solution. In this case, the DM requested to add bounds for SS and SLT because he was only interested in SS values lower than the expected demand and SLT values below four days. He also requested that at least one day's worth of demand for SS or one day SLT be shown. Furthermore, he was only interested in ITO values of at least ten and he preferred to see the probability of product unavailability (PPU) rather than PPA. Therefore, we switched the fifth objective to minimize $PPU = 1 - PPA$.

By running NSGA-III several times, as described in Section 4.2, we generated 1503 nondominated solutions that approximate the Pareto optimal front. We estimated the ideal and nadir points from these solutions and obtained z^* = (358 884.48, 1 000, 674.73, 0.9945, 0, 97.45) and z^{nad} = (367 637.76, 6 800, 4 782.04, 0.5, 0.5, 10.19). After presenting these points to the DM, the navigation process was conducted. The summary of this process can be seen in Table 3. The first column of this table (St) presents the step where the DM provided the reference point as his preferences. Even though the DM was able to stop at any step to change his preferences, in this case, he always let the navigation converge to a single solution. He then analyzed the reachable ranges before deciding to go back to define a new reference point.

The DM initially preferred to set the ideal point as the reference point to investigate how the navigation ran and which Pareto optimal solution he could obtain. From the solution obtained, he observed that, the upper bounds for the reachable values of CSL and ITO were significantly decreased in step 52. Therefore, he decided to return to step 50 to provide new preferences.

After providing new preferences and finding a new solution, the DM found that the CSL value was not satisfactory enough for him. He then decided to go backwards to step 80 since the upper bound of the CSL's reachable values started to decrease at this step. However, after returning two times to step 80 with different preferences, he got the same solution that was not satisfied him. He then

TABLE 3 Summary of the decision making process with NAUTILUS Navigator for item 2

St	Reference point		Solution		The DM's decision
1	PC: 358 884.48 OC: 1 000 HC: 674.73	CSL: 0.9945 PPU: 0 ITO: 97.45	PC: 358 884.48 OC: 4 400 HC: 1 011.40	CSL: 0.7504 PPU: 0.1414 ITO: 47.67	The upper bounds of the reachable CSL and ITO significantly decreased in step 52. Then, go to step 50.
50	PC: 367 637.76 OC: 6 800 HC: 901.98	CSL: 0.9835 PPU: 0 ITO: 59.53	PC: 363 261.12 OC: 6 000 HC: 1 108.19	CSL: 0.8437 PPU: 0.0159 ITO: 39.25	The upper bound of the CSL's reachable values started to decrease at step 80, then return to this step.
80	PC: 367 637.76 OC: 6 800 HC: 901.98	CSL: 0.9835 PPU: 0 ITO: 51.29	PC: 363 261.12 OC: 6 400 HC: 1 141.85	CSL: 0.9161 PPU: 0.0159 ITO: 37.68	Return to step 80 again to provide new preferences in order to improve CSL.
80	PC: 367 637.76 OC: 6 800 HC: 4 038.62	CSL: 0.9945 PPU: 0 ITO: 36.19	PC: 363 261.12 OC: 6 400 HC: 1 141.85	CSL: 0.9161 PPU: 0.0159 ITO: 37.68	No change in the solution, then go back much further to step 16, when the HC started to decrease.
16	PC: 367 637.76 OC: 6 800 HC: 4 782.04	CSL: 0.9945 PPU: 0 ITO: 48	PC: 363 261.12 OC: 6 400 HC: 1 183.94	CSL: 0.9366 PPU: 0.0159 ITO: 35.68	Return to step 75 when the CSL was decreased, and try to relax PPU to get better CSL.
75	PC: 367 637.76 OC: 6 800 HC: 4 782.04	CSL: 0.9945 PPU: 0.5 ITO: 48	PC: 363 261.12 OC: 5 800 HC: 1 066.10	CSL: 0.9272 PPU: 0.1414 ITO: 42.69	Try to put these preferences from the first step to get a better CSL and an acceptable ITO.
1	PC: 367 637.76 OC: 6 800 HC: 4 782.04	CSL: 0.9945 PPU: 0.5 ITO: 40	PC: 367 637.76 OC: 5 800 HC: 1 061.90	CSL: 0.9945 PPU: 0.5 ITO: 42.94	The DM was pleased with the solution.

decided to go further backwards to step 16 because the upper bounds for the reachable values of ITO and HC were significantly decreased after this step.

In step 16, the DM provided new preferences and found some improvement in the CSL value of the solution obtained, but it was not satisfactory enough for him. He realized that CSL had a trade-off with PPU, and then he returned to step 75 when the CSL was decreased, to relax PPU for a better CSL value. However, he did not get a better CSL in the solution obtained, even though he satisfied with the improvement of ITO. He then decided to go to the very first step to set his new preferences. He let the navigation converge to a single solution and was very happy with the solution obtained. He found the CSL value was very good and the other objective function values were acceptable. Therefore, he decided to accept this solution as the final one.

Similar to Case study 1, the Pareto optimality of this solution cannot be guaranteed, because it was generated with an evolutionary algorithm. Therefore, if desired, this solution could be projected onto the Pareto optimal front. However, in this case, the DM was satisfied with the solution obtained and chose to keep it as a final one. From the experience in the previous case study, projecting the solution onto the Pareto optimal front took a significant amount of time in term of several days, because of the expensive functions. Therefore, we did not conduct the projection when it was not necessary for the DM.

Corresponding to the final solution, we had order quantity for each period as well as SS and SLT as the decision variables. The blue line in Figure 3 presents the incoming order quantities for each week, which are the previously set order data for $t = 1, ..., 6$ followed by the optimized order quantities $Q(t - L)$ for $t = 7, ..., 41$. The values of other decision variable were $SS = 92$ and $SLT = 0$. With the same reason as case study 1, we also present demand, inventory level, and safety level in this figure. Based on the inventory level in the green line, the company had excess inventory during the first six weeks, which cannot be controlled by the model due to the lead time. Then, the inventory level decreased and followed the demand quantity to have a higher ITO.

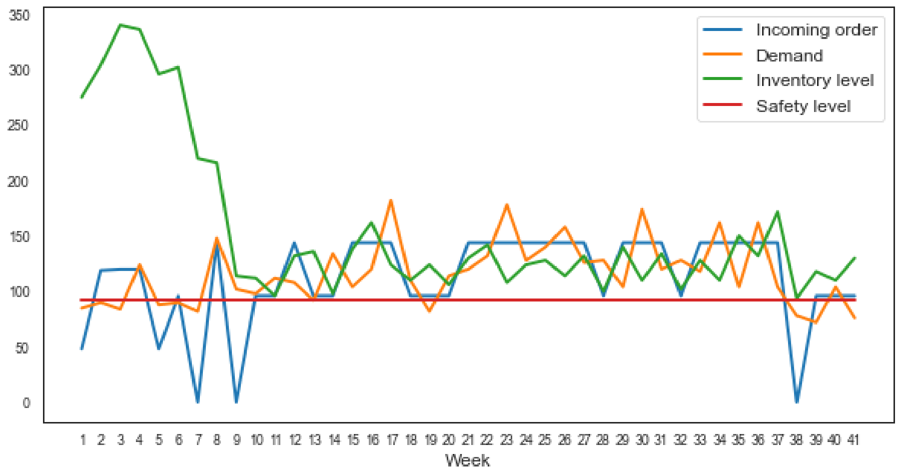


FIGURE 3 Optimized order quantities, demand, inventory level, and safety level for item 2 in the time period considered

4.5 Case study 3: Determining lot sizes with minimum order quantity with hybridization of NAUTILUS Navigator and NIMBUS

In article [PIV], we propose a hybridization of methods combining the strengths of a trade-off-free method NAUTILUS Navigator and a trade-off-based method NIMBUS. As mentioned in Chapter 1, the idea of this hybridization of methods is to use the result from NAUTILUS Navigator as a starting point for NIMBUS. In this case study, we solve the lot sizing problem with MOQ, defined in Section 3.4, for item 3. We use model (12) in this case study, because the DM wanted to obtain lot sizes for each period, simultaneously with MOQ, SS, and SOT. However, the DM requested to see ML values in days, instead of weeks. Thus, we define MOQ DoS (MD), which is calculated by dividing ML by five working days, and we switched the first objective function to minimize MD.

Because of the computationally expensive problem, we generated nondominated solutions as described in Section 4.2. We had 1554 nondominated solutions to be used in both NAUTILUS Navigator and NIMBUS. First, we conducted the decision making process with NAUTILUS Navigator. We presented the estimated ideal and nadir points to the DM, which were derived from the pre-generated set of nondominated solutions. The ideal point was z^* = (1.34, 28 780.4, 400, 237.98, 68.07, 0.9996), and the nadir point was z^{nad} = (41.4, 44 563.2, 3 400, 544.74, 8.67, 0.9058).

The DM started the navigation process by setting all cost objective functions (PC, OC, and HC) to their worst values, CSL to the best value, MD to 5, and ITO to 50. He let the navigation continue until the end and obtained the Pareto optimal solution z = (4.01, 43 866.9, 3 000, 260.07, 43.34, 0.9883). This solution was rather close to his preferences and had a much better value for HC than the desired value that he provided. He was quite satisfied with this solution and wanted to improve it with NIMBUS.

We then conducted the decision making process with NIMBUS with z = (4.01, 43 866.9, 3 000, 260.07, 43.34, 0.9883) as the starting point. We present the summary of the decision making process with NIMBUS in Table 4. The DM provided preference information in each iteration in the form of classification and he wanted to see up to four solutions. The solutions generated based on the DM's preferences are referred to as $z(i, j)$, where i is the iteration number and j is the solution number. The selected solution in each iteration is presented in bold.

After providing his preferences in the first iteration, the DM was presented with four different solutions. He found that they all had the same OC value and he wanted to improve it. He realized that the preferences that he provided were not reachable and, therefore, he decided to continue to the next iteration with the first solution, which was the same as the starting point, to provide different preferences.

In the second iteration, the DM provided different preferences and obtained four different solutions. Among these solutions, the last one was the most satisfactory for him because MD was only a slightly higher than the expected demand for one period. With this solution, he also got lower PC and OC values, and the HC value was acceptable. The CSL and ITO values in this solution were slightly lower than his expectations, but they were still acceptable. The DM was pleased with this solution, but he wanted to try a further iteration to potentially find a better solution. He knew that NIMBUS provides the opportunity to save the solution and, therefore, he could return to this solution if he could not find a better solution in the next iteration.

The DM continued to the third iteration, and obtained four solutions, which

TABLE 4 Summary of the decision making process with NIMBUS for item 3

Iteration	Preference information	Solutions generated
0	starting point	$z(0)$ = (4.01, 43 866.9, 3 000, 260.07, 43.34, 0.9883)
1	MD: impaired until 5 PC: changed freely OC: changed freely HC: changed freely ITO: improved CSL: kept	$z(1,1)$=**(4.01, 43 866.9, 3 000, 260.07, 43.34, 0.9883)** $z(1,2)$=(2.67, 43 170.6, 3 000, 237.98, 68.07, 0.9058) $z(1,3)$=(2.67, 43 170.6, 3 000, 250.03, 51.36, 0.9616) $z(1,4)$=(4.01, 43 170.6, 3 000, 251.37, 50.46, 0.9716)
2	MD: impaired until 5 PC: changed freely OC: improved HC: changed freely ITO: impaired until 35 CSL: impaired until 0.98	$z(2,1)$=(4.01, 41 220.2, 2 200, 280.87, 36.32, 0.9869) $z(2,2)$=(30.72, 31 379, 400, 517.45, 9.65, 0.9979) $z(2,3)$=(8.01, 37 459.3, 1 000, 325.01, 23.25, 0.9744) $z(2,4)$=**(6.68, 37 136, 1 200, 329.12, 27.09, 0.9744)**
3	MD: impaired until 7 PC: changed freely OC: impaired until 1600 HC: changed freely ITO: improved CSL: kept	$z(3,1)$=**(5.34, 43 170.6, 1 600, 307.61, 33.39, 0.9744)** $z(3,2)$=(2.67, 43 170.6, 3 000, 237.98, 68.07, 0.9058) $z(3,3)$=(4.01, 43 170.6, 2 600, 256.73, 48.31, 0.9576) $z(3,4)$=(5.34, 43 170.6, 2 200, 264.76, 44.57, 0.9576)

were very interesting for him, after providing his new preferences. He was pleased with the MD values of the first and the last solutions that were close to what he wanted. However, he did not want to risk his production with the low CSL value in the last solution. Therefore, the first solution aligned perfectly with his preferences, and he decided to stop with it as the final one. Similar to Case study 2, in this case, the DM did not see the need of projecting the solution onto the Pareto optimal front, and wanted to keep this solution to be applied in practice.

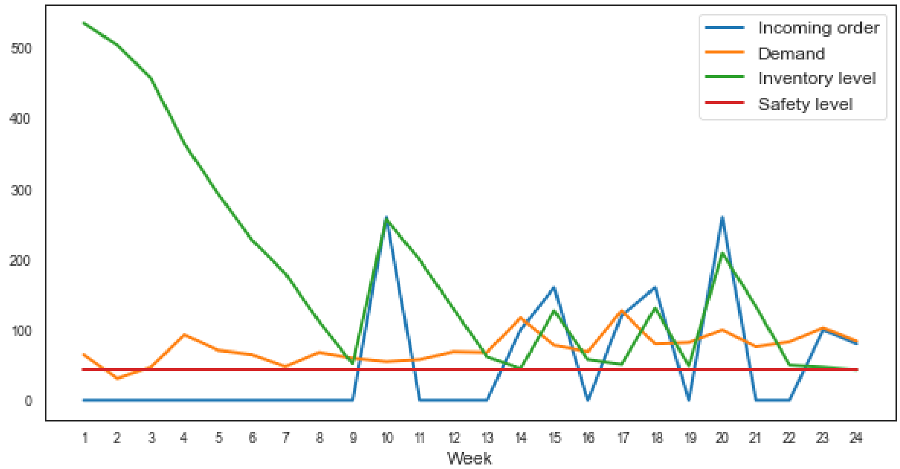


FIGURE 4 Optimized order quantities, demand, inventory level, and safety level for item 3 in the time period considered

The decision variable values corresponding to the final solution were $SS = 43$ units, $SOT = 0$, $MOQ = 80$, and the lot sizes for each period considered presented in Figure 4 in blue. Based on this result, orders are only needed a few times to minimize OC and have a balance among the objective functions, which follows the DM's preferences. The inventory level, which is indicated by the green line in Figure 4, shows that the company had excess inventory at the beginning of the period that could satisfy the demand for nine periods. Similar to the previous case studies, the high inventory level in the beginning of the period could not be controlled by the model, and it then decreased when the model was involved. The DM was pleased with this solution that could support him in negotiations with his supplier.

5 A DECISION SUPPORT APPROACH FOR MULTI-ITEM LOT SIZING

In reality, a supply chain manager of a company needs to solve his/her lot sizing problem not only for a single item but for many items which can be thousands of items for a big company. Conducting a decision making process for every single item as presented in the previous chapter is laborious in this condition. In article [PIV], we propose a decision support approach, called DESMILS, to tackle this challenge. This approach enables any single-item lot sizing model, which is formulated as a multiobjective optimization model, to be used in solving a multi-item problem with a large number of items. In this chapter, we introduce DESMILS, followed by a case study to demonstrate how DESMILS can be used in practice.

5.1 DESMILS

The goal of solving a multiobjective optimization problem is finding a Pareto optimal solution that best represents the DM's preferences. In [PI], we supported the DM to find solutions of model (6) for two items with high and low demands, with different preferences for them. Similarly, in [PIII], the DM expressed different preferences when solving model (10) for two items with high and low MOQ levels to obtain the lowest prices, and we supported him to find his most preferred solutions. From these experiences, we learned that the DM may have different preferences in solving a single-item lot sizing model for different items, but the preferences may be similar for some items that have similarities in some properties. We call the preference information that the DM provides for solving a single-item lot sizing model for a specific item as item-specific preference information.

DESMILS extends a single-item multiobjective lot sizing model to be applied in solving the same lot sizing problem but with a large number of items. Any variant of a single-item lot sizing model can be applied in DESMILS, as long as it is modeled as a multiobjective optimization model. All models proposed in Chapter 3 can be extended to solve the defined lot sizing problems with many items. Before starting with DESMILS, we need to have a single-item lot sizing model to work on and the data to solve it for all items that need to be considered.

DESMILS is designed for solving a multi-item lot sizing problem accommodating item-specific preference information as much as possible without having to conduct the decision making processes for every single item. As said, preference information from the DM can be similar or different depending on the similarities of the items in some properties, such as demand, price, and/or physical size of the item. In DESMILS, m items are divided into c clusters based on these properties, so that items within the same cluster can be assumed to have similar item-specific preference information. Here c is clearly smaller than m, and must be specified by the DM depending on his/her acceptable number of decision making processes that he/she is willing to conduct. Each cluster has one representative item, called a *cluster center*, and the other items, called *cluster members*. Different clusters may have a different number of cluster members, and it is possible that a cluster only has a cluster center without any cluster members.

Assuming that the items in the same cluster can be treated with similar item-specific preference information, the decision making process is only needed to be conducted for one representative item for each cluster, which is the cluster center. Then, for each cluster, the preference information obtained from the DM for the cluster center is transformed to find a reference point for each cluster member in the same cluster. The reference point represents the desired point that the DM wants to achieve for each cluster member, and the solution for each cluster member is derived from this point.

There are four stages in DESMILS that can be described as follows:

1. clustering stage, where m items are divided into c clusters;
2. decision making stage, where the DM conducts c decision making processes to solve the single-item lot sizing model for each cluster center, with an interactive multiobjective optimization method;
3. deriving reference points stage, where, for each cluster member in each cluster, a reference point is derived accommodating preference information from the DM obtained for the corresponding cluster center; and
4. solution generation stage, where solutions for all cluster members in all clusters are generated.

In the first stage, the DM is asked to define the properties that influence his/her opinion in making lot sizing decisions. These properties are important to ensure that items with similar item-specific preference information stay in the same cluster. With these properties, we divide items into clusters by using the k-medoids clustering technique [31]. This technique fits our purpose because it takes an item that is nearest to the means of items as the center of the corresponding cluster. This ensures that the cluster center is an item. Naturally, any other clustering techniques can be used in this stage, as long as it provides one of the items in a cluster as the cluster center and not, for example, some average.

In the second stage, c decision making processes are conducted to solve the single-item lot sizing model for each cluster center as a representative of the rest of the items. Any appropriate interactive multiobjective optimization methods can be utilized to solve the problem, as long as it has a starting point where the DM starts from. The starting point is needed in the next stage to transform the DM's preference information for the cluster center to the preference information for cluster members. In the case study, we used NIMBUS and set a neutral compromise solution as the starting point. After conducting c decision making processes in this stage, for each cluster center, we have a starting point and a final solution as the most preferred solution selected by the DM.

The preference information that the DM provides in solving lot sizing for a cluster center in the second stage is interpreted as the direction from the starting point to the final solution. This direction is called a reference direction, and each cluster center has its own reference direction. In the third stage, for each cluster member, a reference point is derived from the reference direction of the corresponding cluster center. Since the same process is repeated for each cluster, in what follows, we give an example of the solution process for a single cluster.

In solving a single-item lot sizing model for different items, the set of Pareto optimal solutions is also different. This means that the cluster center and its cluster members have different sets of Pareto optimal solutions and different feasible objective regions. Therefore, the reference direction of the cluster center $\boldsymbol{zr}$ needs to be transformed to the feasible objective regions of each cluster member. For this purpose, we first normalize $\boldsymbol{zr}$ to a proportional position $\dot{\boldsymbol{z}}\boldsymbol{r}$, and then denormalize $\dot{\boldsymbol{z}}\boldsymbol{r}$ to each feasible objective region of the cluster member. Each cluster member then has its own reference direction $\boldsymbol{yr}$ as the result of this transformation and it is used to obtain a reference point $\boldsymbol{y}$ for each cluster member.

In order to calculate the reference direction $\boldsymbol{yr}$ for each cluster member, we need a starting point ($\boldsymbol{ys}$). The starting point is calculated in the same way as the interactive multiobjective optimization method in the second stage does (e.g., since we use NIMBUS and a neutral compromise solution as the starting point of the cluster center in the second stage, then we calculate a neutral compromise solution as the starting point for each cluster member). The algorithm to generate a reference point for each cluster member, that we propose, is presented in Algorithm 1, while symbols used in this algorithm can be seen in Table 5.

TABLE 5 List of symbols

Symbol	Description
$\boldsymbol{zs} = (zs_1, \dots, zs_1)$	Starting point of the cluster center
$\boldsymbol{z} = (z_1, \dots, z_1)$	Final solution of the cluster center
$\boldsymbol{zr} = (zr_1, \dots, zr_k)^T$	Reference direction for the cluster center
$\dot{\boldsymbol{z}}\boldsymbol{r} = (\dot{z}r_1, \dots, \dot{z}r_k)^T$	Normalization of $\boldsymbol{zr}$
$\boldsymbol{ys} = (ys_1, \dots, ys_k)^T$	Starting point of the cluster member
$\boldsymbol{yr} = (yr_1, \dots, yr_k)^T$	Reference direction for the cluster member
$\boldsymbol{y} = (y_1, \dots, y_k)^T$	Reference point for the cluster member

Algorithm 1: Algorithm to derive reference points for each cluster member for one cluster

Input: $\boldsymbol{zs}$ and $\boldsymbol{z}$

Output: $\boldsymbol{y}$ for each cluster member

Calculate $\boldsymbol{zr}$ with the following formula:

$$zr_i = z_i - zs_i \ , \ i = 1, \ldots, k.$$

Normalize $\boldsymbol{zr}$ to a proportional position $\dot{\boldsymbol{zr}}$ where:

$$\dot{zr}_i = \frac{zr_i}{zs_i} \ , \ i = 1, \ldots, k.$$

foreach *cluster member* **do**

Calculate a starting point of the cluster member ($\boldsymbol{ys}$).

Calculate $\boldsymbol{yr}$ by denormalizing $\dot{\boldsymbol{zr}}$ into the feasible objective region of cluster member as follows:

$$yr_i = \dot{zr}_i \, ys_i \ , \ i = 1, \ldots, k.$$

Calculate $\boldsymbol{y}$ where:

$$y_i = yr_i + ys_i \ , \ i = 1, \ldots, k.$$

end

From the third stage, each cluster member has its own reference point $\boldsymbol{y}$. However, $\boldsymbol{y}$ may not be a Pareto optimal solution for the lot sizing problem of the cluster member. To find a Pareto optimal solution as the final solution, in the last stage, we minimize ASF (3) with $\boldsymbol{y}$ as the reference point. This process is then repeated for each cluster member to find a final solution for each cluster member. In this way, a Pareto optimal solution that represents the DM's preferences is found for each item considered.

5.2 Case study

A case study was presented in article [PIV] to demonstrate the applicability of DESMILS. In this case, the supply chain manager as the DM needed to solve a lot sizing problem with demand uncertainty, and model (6) was best suited for this purpose. Therefore, in this section, we demonstrate how DESMILS extends model (6) to be used in solving a multi-item lot sizing problem with uncertainty on demand, but we present a different cluster.

There were 94 items that the DM needed to consider, and he was only able to conduct the decision making processes for 7 to 12 items. The time period for inventory planning was one week, and there were 24 weekly periods that he considered. Based on our discussion with the DM, it turned out that there are six relevant properties that influence his decisions in lot sizing. They are: SS, SOT, price, daily demand, transit time, and physical size of an item. The DM then provided the data of these six relevant properties for 94 items from the ERP system of the company. The data needed for solving lot sizing model (6), which is described in Section 3.2, for 94 items, is also provided from the same source.

5.2.1 Clustering stage

We used the k-medoids clustering technique in the first stage to divide items into clusters and set the number of clusters from 7 to 12 clusters, as the acceptable number of decision making processes provided by the DM. The DM then saw and compared the clustering results (i.e, cluster centers and cluster members for different numbers of clusters). He found that the result with 10 clusters was the best for him, because, with this result, the items in the same cluster could be treated with similar preferences. This clustering result of 10 clusters is presented in Figure 5.

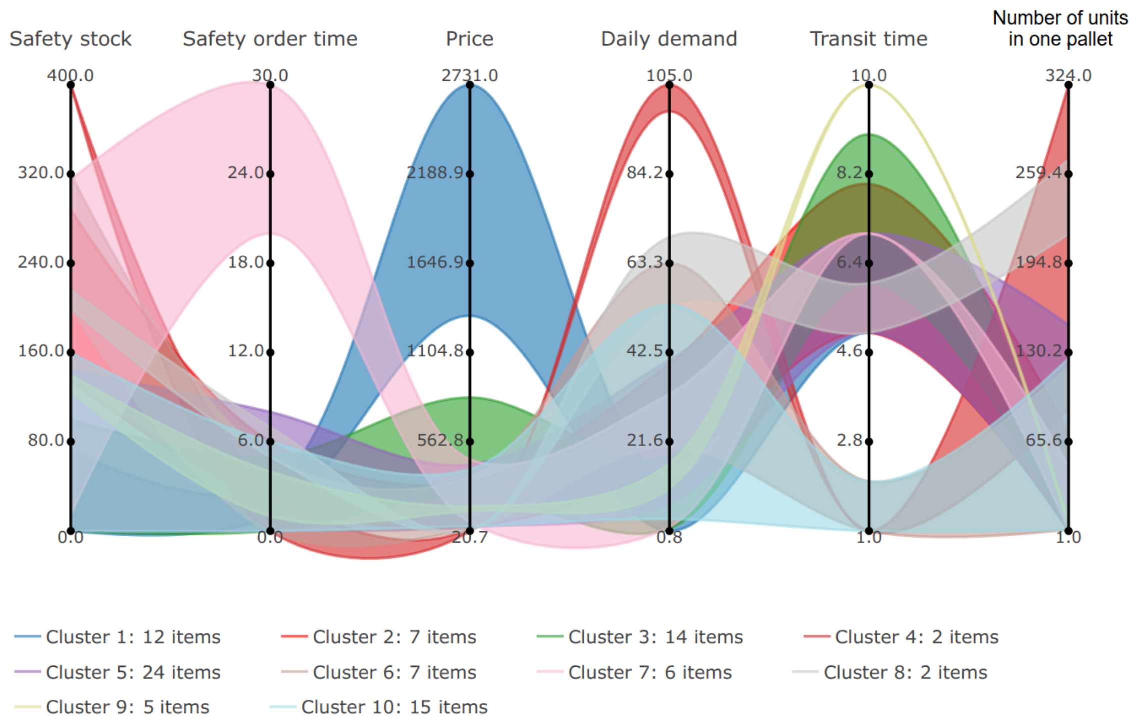


FIGURE 5 Result of clustering, where ten clusters are indicated by different colors

5.2.2 Decision making stage

In this stage, the DM conducted ten decision making processes to solve the lot sizing model (6) for each cluster center. For compactness, here, we describe the decision making process for one cluster center only, while the remaining ones were treated in the same way. Different from the one presented in article [PIV], in this thesis, we consider the cluster having items of high prices and low daily demands. This cluster is shown in blue color in Figure 5 (cluster 1).

The data provided by the company showed that the cluster center of this cluster has a price of €1890 and a lead time of five weeks. This item has a minimum order quantity of 2 units, and the rounding value is the same. The demand data and the previous orders supposed to arrive during the lead time period of this item are presented in Figure 6.

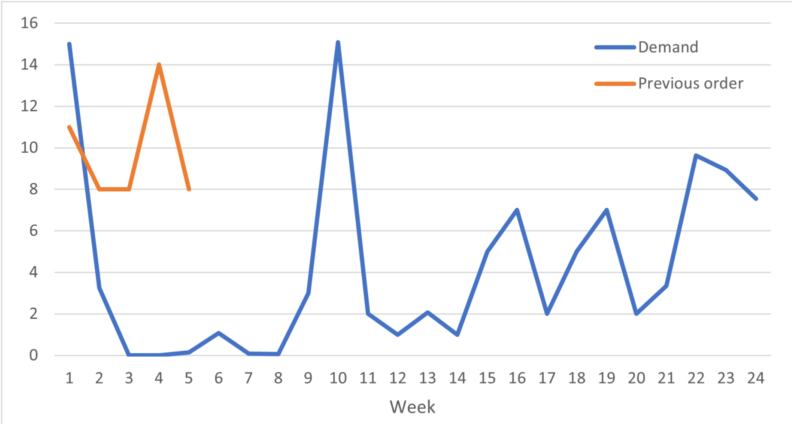


FIGURE 6 Demand and previous order data for the cluster center

We used the synchronous NIMBUS method to find the best lot sizes for the cluster center simultaneously with SS and SOT. As described in Section 4.2, lot sizing problems are computationally expensive, and therefore, we generated a representative set that approximates Pareto optimal solutions in advance to reduce the waiting time of the DM. NIMBUS was then used to help the DM select one of them as a final solution. This representative set is generated in the same way as we did in the case studies described in the previous chapter.

By using the representative set, the DM conducted the decision making process with NIMBUS. Before providing the first preferences in the form of classification, the neutral compromise solution as a starting point was presented to the DM, together with the ideal and nadir points. Table 6 presents a summary of the decision making process for this item. In each iteration, the DM wanted to compare up to four solutions. However, in iterations 2 and 3, he only got three different solutions because two scalarizing functions resulted in the same solution. The selected solution in each iteration is presented in bold.

As presented in Table 6, in the first iteration, the DM was not really satisfied with the CSL value of the starting point. He wanted to improve CSL until 0.95 and allowed ITO to decrease until 40, while POC and HC were allowed to change freely. With this classification, he got four different solutions, and he preferred the third solution that gave him a quite good CSL value, but the ITO value was still acceptable. The DM was already rather satisfied with the current solution but he decided to continue to the next iteration with this solution to explore whether he could get a better one.

In the second iteration, the DM wanted to investigate what solutions he could get if he wanted to improve CSL as much as possible. To get that high CSL, he could sacrifice ITO until the worst value while POC and HC were still allowed to change freely. He found three different solutions with this classification and selected the second one that had the best ITO to continue to the next iteration.

TABLE 6 Summary of the decision making process with NIMBUS for the cluster center

Iteration	**Preference information**	**Solutions generated**
0	starting point	$z(0)$ = (77 000, 1 988.63, 0.5891, 74.57)
1	POC: changed freely HC: changed freely CSL: improved until 0.95 ITO: impaired until 40	$z(1,1)$=(84 560, 2 048.77, 0.8163, 40.93) $z(1,2)$=(92 520, 2 186.22, 0.9643, 28.2) $z(1,3)$=**(88 540, 2 130.38, 0.9426, 31.35)** $z(1,4)$=(84 560, 2 065.95, 0.87, 37.81)
2	POC: changed freely HC: changed freely CSL: improved ITO: impaired until 20	$z(2,1)$=**(92 520, 2 186.22, 0.9643, 28.2)** $z(2,2)$=(91 520, 2 306.49, 0.9642, 24.15) $z(2,3)$=(92 320, 2 194.81, 0.9642, 27.93)
3	POC: changed freely HC: changed freely CSL: impaired until 0.95 ITO: improved until 50	$z(3,1)$=**(92 520, 2 186.22, 0.9643, 28.2)** $z(3,2)$=(88 540, 2 113.2, 0.9118, 32.99) $z(3,3)$=(84 560, 2 048.77, 0.8163, 40.93)

In the third iteration, he allowed to impair CSL until 0.95 but wanted to improve ITO until 50. He let the other objective functions change freely. After comparing the solutions obtained, he preferred the first one, which was the same solution as in the previous iteration. The DM was satisfied with this solution and decided to stop with this as a final one.

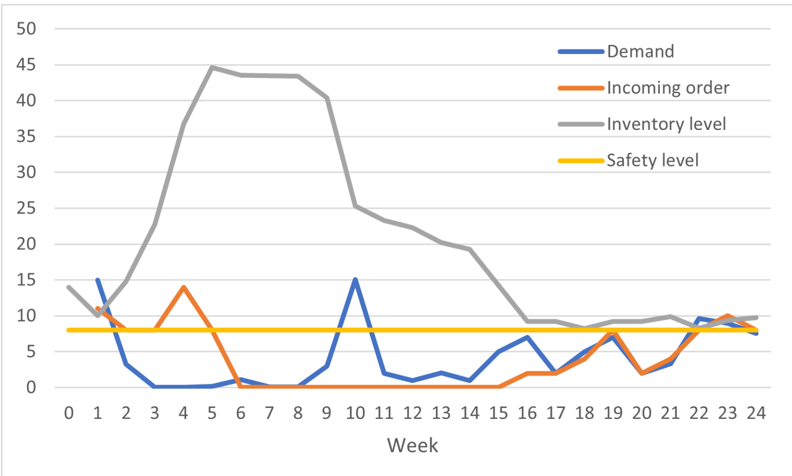


FIGURE 7 Result for cluster center

Corresponding to the final solution, we had lot sizes for each period considered, as well as SS and SOT, as decision variables. Figure 7 presents the lot sizes that arrive for each period in orange, where the first five weeks are the previously set order data followed by the optimized lot sizes after week 5 (note that the lead time was 5 weeks for this item). The final value of SS was 8 days and the SOT value was zero. The inventory level indicated by the grey line shows that the company had excess inventory due to the previous orders that could not be controlled by the model. Therefore, the optimization result suggested no order until week 15. Then, with the optimized lot sizes, the inventory level decreased but was still sufficient to keep the safety level high, following the DM's preferences. The DM was pleased with this improvement in the inventory level, which could save money invested in the inventory with high safety level.

5.2.3 Deriving reference points stage

In the previous stage, the DM directed his preferences with NIMBUS from the starting point $\boldsymbol{zs}$ = (77 000, 1 988.63, 0.5891, 74.57) towards his most preferred solution $\boldsymbol{z}$ = (92 520, 2 186.22, 0.9643, 28.2) for the cluster center. Using this information, we calculated the reference direction of the cluster center as $\boldsymbol{zr}$ = (15 520, 197.59, 0.3752, -46.37), and then normalized it to obtain $\dot{\boldsymbol{zr}}$ = (0.2016, 0.0994, 0.6368, -0.6218).

TABLE 7 Starting points and reference points for cluster members

Item	Starting points				Reference points			
	POC	**HC**	**CSL**	**ITO**	**POC**	**HC**	**CSL**	**ITO**
1	1 015 607	18 622.84	0.9075	11.02	1 078 310.2	8 242.28	0.9284	18.04
2	878 238	17 083.16	0.9186	11.04	932 460.18	7 560.83	0.9397	18.07
3	181 000	4 099.86	0.9	9.51	192 174.89	1 814.56	0.9207	15.57
4	647 428	9 628.77	0.9104	12.06	687 400.03	4 261.6	0.9313	19.73
5	782 222	12 390.21	0.9089	11.18	830 516.17	5 483.78	0.9298	18.3
6	639 654	11 123.38	0.9013	11.71	679 146.06	4 923.1	0.922	19.16
7	224 175	5 394.66	0.9079	7.86	238 015.5	2 387.62	0.9288	12.86
8	1 523 108	14 232.04	0.9021	17.29	1 617 144.3	6 298.96	0.9228	28.3
9	1 056 601	23 260.21	0.9002	9.96	1 121 835.3	10 294.7	0.9209	16.3
10	154 529.5	4 732.22	0.9043	11.39	164070.11	2 094.43	0.9251	18.64
11	1 593 158	26 020.07	0.9156	13.13	1 691 518.9	11 516.2	0.9366	21.49

There were 11 cluster members in this cluster, excluding the cluster center. Following Algorithm 1, we needed to calculate starting points for these 11 cluster members. Since we utilized NIMBUS and the neutral compromise solution as the starting point for the cluster center, therefore, for each cluster member, we also calculated a neutral compromise solution as the starting point. We then followed the algorithm to obtain a reference point for each cluster member. The starting and the reference points for each cluster member in this cluster are summarized in Table 7.

5.2.4 Solution generation stage

From the reference points obtained in the previous stage, in this stage, we minimized the ASF (3) to derive a solution for each cluster member. These solutions are listed in Table 8. The DM accepted these solutions as well as the corresponding lot sizes, SS, and SOT as the decision variables for each item. He appreciated that he was able to obtain different solutions for 12 items in this cluster representing his preferences, with only one decision making process.

The processes from the decision making stage to the solution generation

TABLE 8 Solutions for cluster members

Item	POC	HC	CSL	ITO
1	1 029 606.96	7 748	0.9351	45.6
2	884 692	7 107.21	0.9483	54.1
3	188 560	1 678.18	0.9455	26.4
4	654 480	2 624.34	0.9497	47.32
5	789 416	2 687.8	0.9453	50.64
6	647 716	3 395.89	0.9425	43.07
7	226 730	2 048.73	0.9355	24.91
8	1 531 700	5 013.21	0.9343	59.73
9	1 062 384.4	7 747.58	0.9328	69.68
10	159 675	3 205.93	0.9363	27.44
11	1 598 142.8	12 783.94	0.9359	70.94

stage was repeated for the remaining clusters. In the decision making stage, the DM provided different preferences for the different cluster centers. He checked all the solutions obtained with DESMILS, and he was satisfied with all solutions and the corresponding decision variables of both cluster centers and cluster members, which followed his preferences. The DM appreciated the benefit of DESMILS to find solutions for 94 items that best represent his preferences, with only 10 decision making processes.

6 CONCLUSIONS AND AUTHOR'S CONTRIBUTIONS

In the final chapter of this thesis, we provide conclusions and future research directions. Then, we elaborate on the author's contributions in the included articles [PI]-[PIV], followed by final thoughts at the end of this chapter.

6.1 Conclusions

The main focus of this thesis is to model and solve lot sizing problems inspired by real challenges. We first focused on single-item lot sizing problems in three various conditions. We then proposed three multiobjective optimization models to address these challenges and solve them with appropriate interactive multiobjective optimization methods. Furthermore, we proposed DESMILS to extend the application of these single-item models to be used to solve multi-item problems that accommodate different preferences from the DM for different items without having to conduct the decision making process separately for every single item.

In article [PI], we developed a multiobjective optimization model to address the challenge of solving a single-item lot sizing problem under a stochastic environment on demand. We introduced a new and practically viable way to handle unpredicted demand, called safety order time, that can handle high fluctuations of demand better than safety stock. The proposed multiobjective optimization model had four objective functions, where both safety stock and safety order time were used to increase the preparedness of handling demand uncertainty. By using the proposed model, the optimal order quantities in the periods considered were simultaneously determined with the optimal values of safety stock and safety order time.

In article [PII], we considered a single-item lot sizing problem under a stochastic environment on demand and lead time. We used a safety stock to handle uncertainty on demand and a cycle service level formula to measure the quality of safety stock. To handle uncertainty on the lead time, we used a safety lead time, and we proposed the probability of product availability formula to measure the quality of safety lead time. We integrated the lot sizing problem with the problem of determining the optimal values of safety stock and safety lead time by proposing a multiobjective optimization model. This model contains six objective functions, including cycle service level and probability of product availability, to find the optimal order quantity in each period, as well as the optimal values of safety stock and safety lead time.

Furthermore, in article [PIII], we integrated a single-item lot sizing problem with the problem of determining a minimum order quantity to support a company as a buyer. We developed a multiobjective optimization model with five objective functions and proposed a minimum order quantity level formula as one of the objective functions to assess the quality of minimum order quantity in satisfying demand. By solving the proposed model, the DM could simultaneously determine an optimal minimum order quantity with the optimal lot sizes for each period. The insight gained in the decision making process can be used in a negotiation with a supplier related to a minimum order quantity.

Real data from a manufacturing company was utilized to demonstrate the applicability and usefulness of the proposed lot sizing models and concepts. A supply chain manager from the said company acted as the DM to provide his domain expertise in the decision making processes. We applied interactive methods to solve the problems since these methods had many benefits in supporting the DM to determine his most preferred solution. We used different interactive methods to solve the proposed models based on the wishes of the DM and the models to be solved. Before the decision making processes, we generated a set of nondominated solutions approximating Pareto optimal solutions to avoid introducing waiting times on the DM because of lengthy computation times. We applied the E-NAUTILUS method in article [PI] and the NAUTILUS Navigator method in article [PII]. They allowed the DM to find preferred solutions without thinking of sacrifices and trade-offs. In article [PIII], we proposed to apply a hybridization of methods combining the strengths of NAUTILUS Navigator as a trade-off-free method and NIMBUS as a trade-off based method.

We then extended the research in article [PIV] by proposing DESMILS to enable the single-item lot sizing models from articles [PI]-[PIII], as well as other single-item multiobjective lot sizing models, to be applied in multi-item problems with a large number of items. This approach solves the single-item lot sizing model with an interactive method only for few selected items. The preferences obtained from the DM for these selected items are then used to derive optimal lot sizes for the other items that can be treated with similar preferences in the lot sizing decision. In this way, optimal lot sizes that represent the DM's preferences are obtained for all items without the need for the DM to repeat the decision making process for each item separately. By using real data from a manufacturing company, we demonstrated how, with DESMILS, Pareto optimal solutions reflecting the DM's preferences could be found for 94 items, and the DM only needed to solve the lot sizing problem for 10 selected items.

From the decision making processes and the results reported in this thesis, we can conclude that the DM appreciated the benefit of using multiobjective optimization in solving real-world lot sizing problems. He found all the concepts, models, interactive solution processes, and results useful in his daily operations. He was also satisfied with all the solutions as well as the corresponding decision variables. To be more specific, the DM highlighted the benefits and insight that he gained from the studies included in this thesis as follows:

1. Multiobjective optimization allows the DM to consider different important indicators as objective functions simultaneously and understand the trade-offs among them. He gained valuable insight from the proposed models, compared with the previous way, where he was only able to optimize one objective function and set others as constraints.

2. During the interactive decision making processes, he was able to compare different solutions reflecting different preferences that allowed him to gain insights into the problem and obtain his most preferred solution. Thus, he could train his team members and other stakeholders of the company about benefits of simultaneously considering multiple objective functions in his problem.

3. He appreciated that the solutions obtained in this thesis improved inventory planning and control in his company. It is presented in the case study that inventory value, which is a critical indicator for top management, was reduced for all items considered.

4. Time saving is a crucial issue in daily operations. Compared to the previous way, where he needed to determine lot sizes item by item, DESMILS significantly reduced his time and effort. In DESMILS, the number of decision making processes to conduct is decided by the DM, and therefore, he could control the effort needed to solve the problem for all items that he had.

5. DESMILS also reduces the risk of human error. When processes are not controlled only by humans, the risk of unintentional forgetting is reduced. It in turn supports production needs when the right amount of items is available at the right time.

There are still several challenges for future research from this thesis. For example, we assumed that demand data follows a normal distribution. One could explore different distributions based on the data and modify the models to adapt them. Furthermore, we also assumed that the cost to place one order is static with no backorder cost. Relaxing assumptions by considering a backorder cost or having a dynamic ordering cost are other possible future research topics.

DESMILS is the first approach that can solve multi-item lot sizing problems incorporating a DM's preferences in deciding lot sizes for different items. Thus, testing this approach with different types and characteristics of the problems and with different numbers of items are also topics of future research extending this study. Furthermore, in this study, we did not consider any information about connectivity and dependency among items, which presents another potential direction for future research. Lastly, our focus in this thesis was on supporting a single DM. Supporting multiple DMs who may have conflicting preferences is a future research direction.

6.2 Author's Contribution

The author was interested in the application of multiobjective optimization to solve real-world problems since the beginning of her doctoral studies. The choice of lot sizing was suggested by her supervisor, Prof. Kaisa Miettinen, who then introduced her to Juha Sipilä, a senior lecturer in production management and materials management, and Jussi Lehtimäki, a supply chain project manager in a manufacturing company. The challenges considered in this thesis were motivated by real-world problems that were brought up in the discussions with them. In the beginning, the author did not have any experience in lot sizing. Thus, the author conducted a literature survey to learn the background concepts of lot sizing and the connection of this problem domain to multiobjective optimization. She also studied in the literature what the typical objective functions in lot sizing are so that they could be utilized together with the models developed by her. Besides, she developed her understanding of lot sizing problems by having more discussions with Juha Sipilä and Jussi Lehtimäki.

In article [PI], the idea of the safety order time concept came from Jussi Lehtimäki, based on a real concept in his company. The author proposed the first version of the multiobjective optimization model, and it was then modified in the discussions with Jussi Lehtimäki to be able to adapt the safety order time in a better way. The idea of considering inventory turnover as one of the objective functions came from Juha Sipilä and was agreed upon by Jussi Lehtimäki, as the DM, since it was an important indicator in his company. Giovanni Misitano developed the graphical user interface for the E-NAUTILUS method in this paper and made it available online. He also contributed in writing about it and editing this article.

The main idea of the probability of product availability formula as well as the model proposed in article [PII], came from the author. This idea arose because of the different understanding of the term 'safety time' in the discussion of article [PI]. This term meant safety order time for Jussi Lehtimäki in his daily operations, but it referred to safety lead time for the author, based on her literature survey. After this, the author initiated to continue study about safety lead time to solve the lot sizing problem with uncertainty on demand and lead time. The idea of dividing purchasing cost and ordering cost into different objective functions came in the decision making process, when Jussi Lehtimäki, as the DM, found difficult to recognize whether the high purchasing and ordering cost objective function was caused by purchasing cost or ordering cost.

The integration problem of lot sizing and the minimum order quantity determination problem in article [PIII] was originally requested by Jussi Lehtimäki based on real needs in his company. The author studied this problem from the literature and got the idea of the proposed minimum order quantity level formula and the corresponding lot sizing model. The idea of hybridization of methods to solve this problem was initiated by Prof. Kaisa Miettinen, after Jussi Lehtimäki, as the DM, found difficulty in finding his most preferred solution by using one method. The author then matured the idea by experimenting with real data.

The idea of creating the decision support approach for multi-item lot sizing problems in article [PIV] arose in the discussions with Jussi Lehtimäki. He has thousands of items in his company, and it is impossible to do the decision making processes for every single item. The author then tested different ideas and experimented with several methods to develop an approach that can accommodate different preferences from the DM in lot sizing decisions for different items. Discussions and suggestions from her supervisors, Prof. Kaisa Miettinen and Dr. Bekir Afsar, helped her develop DESMILS.

The author is the main contributor in all the included articles. She is the one who formulated the problems and developed the proposed models. She then implemented all the models by experimenting with several interactive methods and analyzed the results. In some cases, good results were challenging to be obtained, and some iterations of the processes were needed. She needed to modify the problem formulations and the models, explore other methods, and find appropriate solvers that are applicable to solve these problems. She applied modules available in DESDEO, but this needed a lot of work since she had to tailor the modules to the specific needs of the DM. Her supervisors, Prof. Kaisa Miettinen and Dr. Bekir Afsar helped her very much in all the processes during the regular supervision meeting by giving valuable suggestions.

The author wrote most parts of the included articles. Her supervisors edited her writing and gave valuable comments to improve the writing. Juha Sipilä helped the author in analyzing the results from a practical point of view and wrote few paragraphs in the articles, while Jussi Lehtimäki shared his domain expertise and acted as the DM.

6.3 Final Thoughts

Inspired by real challenges in a manufacturing company, we believe that the studies in this thesis are useful for other companies as well. Those who have similar problems can easily apply the proposed models with their own data. Furthermore, DESMILS can be implemented for any single-item multiobjective lot sizing models, therefore it can be a valuable tool for those who want to optimize their lot sizes for a large number of items.

The aim of this thesis is to explore the benefits provided by multiobjective optimization and interactive methods in solving real lot sizing problems. Throughout this thesis, we have said that the DM gained benefits, not only with the concepts, models, and solutions reported in this thesis, but gained insights while learning about the problem during the interactive decision making processes. We hope that this thesis will provide a guidance to anyone who wishes to use interactive multiobjective optimization in other real-world lot sizing problems. Furthermore, we also plan to make all codes we developed openly available online. We hope that this will increase the impact of this thesis and facilitate the replication and extension of our work by others.

To be more general, by bridging the gap between theory and practice, we emphasize the need of research to address real-world problems. We believe that conducting research on practical problems provides valuable insights for both research and practice. Our hope is that our findings will inspire more researchers to take on these challenges and contribute to provide a greater impact in the practical life.

YHTEENVETO (SUMMARY IN FINNISH)

Toimituserän mitoitusongelmissa on yleensä useita ristiriitaisia tavoitefunktioita, joita tulisi optimoida samanaikaisesti. Täten, näiden ongelmien käsitteleminen monitavoiteoptimointimallien avulla ja niiden ratkaiseminen soveltuvilla menetelmillä on tärkeää, jotta löydetään käytäntöön soveltuvia optimiratkaisuja. Tässä väitöskirjassa käsiteltiin haasteita, joita ilmenee käytännön toimituserän mitoitusongelmissa eri tilanteissa. Työssä tarkasteltiin sekä yhden että useamman nimikkeen toimituserien mitoitusta. Yhden nimikkeen toimituseriin liittyen käsiteltiin kolmea tilannetta: kun kysyntä on epävarmaa, kun sekä kysyntä että toimitusaika ovat epävarmoja ja kun halutaan määrittää minimitoimituserän suuruus. Lisäksi esiteltiin päätöksenteon tukimenetelmä DESMILS usean nimikkeen toimituserien mitoittamiseen.

Jokaiseen käsiteltyyn tilanteeseen esiteltiin monitavoitteinen toimituserän mitoitusmalli, joissa oli mukana uusia muotoiluja ja käsitteitä ja täten täytettiin kirjallisuudessa olevia aukkoja. Ensin esiteltiin varmuusajan käsite kysynnän epävarmuuden kattamiseksi. Toiseksi esiteltiin uusi nimikkeen varastoriittävyyden todennäköisyyttä kuvastava kaava, joka määrittää turvallisen toimitusajan käsitteen käsiteltäessä toimitusajan epävarmuutta. Kolmanneksi esiteltiin kaava minimitoimituserälle, jotta tilaus vastaa tarvetta. Lisäksi DESMILS-menetelmän avulla näitä yhden nimikkeen toimitusmäärän mitoitusmalleja voidaan soveltaa suurten nimikemäärien käsittelyyn. Tämän menetelmän ansiosta riittää, että päätöksentekijä mitoittaa toimitusmäärät interaktiivisella menetelmällä vain muutamalle valitulle nimikkeelle ja muiden nimikkeiden toimitusmäärät päätellään päätöksentekijän antamien mieltymysten avulla.

Esiteltyjen mallien ja käsitteiden soveltuvuutta ja käyttökelpoisuutta havainnollistettiin käyttäen todellista teollisuusyrityksen dataa. Kyseisen yrityksen toimitusketjun johtaja käytti päätöksentekijänä omaa asiantuntemustaan ratkaisuprosesseissa. Häntä tuettiin löytämään parhaat ratkaisut kuhunkin käsiteltyyn ongelmaan hyödyntäen interaktiivisia menetelmiä kuten E-NAUTILUS, NAUTILUS Navigator ja esitettyä kahden menetelmän yhdistelmää. Lisäksi DESMILS-menetelmän avulla päätöksentekijää tuettiin määrittämään Pareto-optimaaliset toimituserät 94 nimikkeelle niin, että ne kuvastavat hänen esittämiään mieltymyksiä yrityksen materiaalinohjauksen tavoitteisiin liittyen. Menetelmän ansiosta hän pystyi tähän ratkaisemalla yhden nimikkeen toimitusmäärän optimointiongelma vain 10 valitulle nimikkeelle.

Päätöksentekijä arvosti monitavoiteoptimoinnin hyötyjä todellisten toimitusmäärän mitoitusongelmien ratkaisemisessa. Hänen mielestään käytetyt käsitteet, mallit, interaktiiviset ratkaisuprosessit ja väitöskirjassa esitetyt tulokset olivat hyödyllisiä hänen päivittäisiä toimintojaan ajatellen. Monitavoiteoptimoinnin avulla hän pystyi käsittelemään useita tärkeitä indikaattoreita tavoitefunktioina samanaikaisesti sekä ymmärtämään niiden välisiä riippuvuuksia ja vaihtosuhteita. Interaktiivisten päätöksentekoprosessien aikana hän pystyi vertailemaan erilaisia ratkaisuja, jotka kuvastivat hänen mieltymyksiään. Tämä lisäsi hänen ymmärrystään toimituserän mitoitusongelmista, ja auttoi löytämään hänelle parhaat ratkaisut. Hän arvosti väitöskirjassa saatuja tuloksia, jotka paransivat yrityksen varaston suunnittelua ja hallintaa. Menetelmät säästivät merkittävästi hänen varaston suunnitteluun ja ylläpitoon käyttämäänsä aikaa ja vaivaa.

REFERENCES

[1] B. Afsar, K. Miettinen, and F. Ruiz. Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys*, 54(4):1–27, 2021.

[2] B. Afsar, J. Silvennoinen, and K. Miettinen. A systematic way of structuring real-world multiobjective optimization problems. In M. Emmerich, A. Deutz, H. Wang, A.V. Kononova, B. Naujoks, K. Li, K. Miettinen, and I. Yevseyeva, editors, *Evolutionary Multi-Criterion Optimization, EMO 2023, Proceedings*, pages 593–605. Springer, Cham, 2023.

[3] D. Alem, E. Curcio, P. Amorim, and B. Almada-Lobo. 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, 2018.

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

[5] T. Amornpetchkul. Threshold discounts comparison: All-unit or incremental? *Journal of Revenue and Pricing Management*, 16:265–294, 2017.

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

[7] T. Aslam and A.H.C. Ng. Multi-objective optimization for supply chain management: A literature review and new development. In *Proceedings of the 8th International Conference on Supply Chain Management and Information Systems*, pages 425–432, Hong Kong, 2010. IEEE.

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

[9] N. Açıkgöz, G. Çağıl, and Y. Uyaroğlu. The experimental analysis on safety stock effect of chaotic supply chain attractor. *Computers & Industrial Engineering*, 150:106881, 2020.

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

[11] G.R. Bitran and H.H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1982.

[12] J. Blank and K. Deb. Pymoo: Multi-objective optimization in Python. *IEEE Access*, 8:89497–89509, 2020.

[13] N. Brahimi, N. Absi, S. Dauzère-Pérès, and A. Nordli. Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3):838–863, 2017.

[14] J.T. Buchanan and J. Corner. The effects of anchoring in interactive MCDM solution methods. *Computers & Operations Research*, 24(10):907—918, 1997.

[15] S. Chopra and P. Meindl. *Supply Chain Management: Strategy, Planning, and Operation*. Pearson, 6 edition, 2016.

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

[17] K. Deb and H. Jain. 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, 2014.

[18] K. Deb, K. Miettinen, and S. Chaudhuri. Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, 14(6):821–841, 2010.

[19] C.H. Glock, E.H. Grosse, and J.M. Ries. The lot sizing problem: A tertiary study. *International Journal of Production Economics*, 155:39–51, 2014.

[20] J.N.C. Gonçalves, M.S. Carvalho, and P. Cortez. Operations research models and methods for safety stock determination: A review. *Operations Research Perspectives*, 7:100164, 2020.

[21] H.G. Goren, S. Tunali, and R. Jans. A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing*, 21:575–590, 2010.

[22] S.C. Graves. Safety stocks in manufacturing systems. *Journal of Manufacturing and Operations Management*, 1:67–101, 1988.

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

[24] Y.Y. Haimes, L.S. Lasdon, and D.A. Wismer. 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, 1971.

[25] F.W. Harris. How many parts to make at once. *Factory, The Magazine of Management*, 10:135–136, 1913.

[26] M. Hartikainen, K. Miettinen, and K. Klamroth. Interactive Nonconvex Pareto Navigator for multiobjective optimization. *European Journal of Operational Research*, 275(1):238–251, 2019.

[27] R. Heikkinen, J. Sipilä, V. Ojalehto, and K. Miettinen. Flexible data driven inventory management with interactive multiobjective lot size optimization. *International Journal of Logistics Systems and Management*. (to appear). doi: 10.1504/IJLSM.2021.10042370.

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

[29] R. Jans and Z. Degraeve. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, 177(3):1855–1875, 2007.

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

[31] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc, 1990.

[32] G. Kirlik and S. Sayın. Computing the nadir point for multiobjective discrete optimization problems. *Journal of Global Optimization*, 62:79–99, 2015.

[33] P. Korhonen and J. Wallenius. Behavioural issues in MCDM: Neglected research questions. *Journal of Multi-Criteria Decision Analysis*, 5(3):178–182, 1996.

[34] A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[35] O. I. Larichev. Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1(3):127–138, 1992.

[36] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[37] K. Miettinen. Introduction to multiobjective optimization: Noninteractive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, volume 5252, pages 1–26. Springer, Berlin, Heidelberg, 2008.

[38] K. Miettinen, J. Hakanen, and D. Podkopaev. Interactive nonlinear multiobjective optimization methods. In S. Greco and J. Figueira M. Ehrgott, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 233, pages 927–976. Springer, New York, 2016.

[39] K. Miettinen and M.M. Mäkelä. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24:193–213, 2002.

[40] K. Miettinen and M.M. Mäkelä. Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3):909–922, 2006.

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

[42] K. Miettinen, F. Ruiz, and A.P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Słowiński, editors, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, volume 5252, pages 27–57. Springer, Berlin, Heidelberg, 2008.

[43] G. Misitano, B. S. Saini, B. Afsar, B. Shavazipour, and K. Miettinen. DESDEO: The modular and open source framework for interactive multiobjective optimization. *IEEE Access*, 9:148277–148295, 2021.

[44] C. New. Safety stocks for requirements planning. *Production and Inventory Management*, 12:1–18, 1975.

[45] Y. Nikulin, K. Miettinen, and M.M. Mäkelä. A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34:69–87, 2012.

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

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

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

[49] A.B. Ruiz, F. Ruiz, K. Miettinen, L. Delgado-Antequera, and V. Ojalehto. NAUTILUS Navigator: free search interactive multiobjective optimization without trading-off. *Journal of Global Optimization*, 74:213–231, 2019.

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

[51] H. Ruotsalainen, K. Miettinen, J.-E. Palmgren, and T. Lahtinen. Interactive multiobjective optimization for anatomy-based three-dimensional HDR brachytherapy. *Physics in Medicine & Biology*, 55(16):4703–4719, 2010.

[52] G. Saccani, J. Hakanen, K. Sindhya, V. Ojalehto, M. Hartikainen, M. Antonelli, and K. Miettinen. Potential of interactive multiobjective optimization in supporting the design of a groundwater biodenitrification process. *Journal of Environmental Management*, 254:109770, 2020.

[53] M. Schmidt, W. Hartmann, and P. Nyhuis. Simulation based comparison of safety-stock calculation methods. *CIRP Annals*, 61(1):403–406, 2012.

[54] E.A. Silver, D.F. Pyke, and D.J. Thomas. *Inventory and Production Management in Supply Chains*. CRC Press, 4th edition, 2017.

[55] K. Sindhya, A. Manninen, K. Miettinen, and J. Pippuri. Design of a permanent magnet synchronous generator using interactive multiobjective optimization. *IEEE Transactions on Industrial Electronics*, 64(12):9776–9783, 2017.

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

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

[58] H.M. Wagner and T.M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89–96, 1958.

[59] D.C. Whybark and J.G. Williams. Material requirements planning under uncertainty. *Decision Science*, 7(4):595–606, 1976.

[60] A.P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making Theory and Application*, pages 468–486. Springer, Berlin, Heidelberg, 1980.

[61] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu. Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access*, 6:41256–41279, 2018.

[62] T. Tanino Y. Sawaragi, H. Nakayama. *Theory of Multiobjective Optimization*. Elsevier, 1985.

[63] P.H. Zipkin. *Foundations of Inventory Management*. McGraw-Hill Higher Education, 2000.

.

ORIGINAL PAPERS

PI

INTEGRATION OF LOT SIZING AND SAFETY STRATEGY PLACEMENT USING INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

by

Adhe Kania, Juha Sipilä, Giovanni Misitano, Kaisa Miettinen, and Jussi Lehtimäki 2022

Computers & Industrial Engineering, vol 173, article no. 108731

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:

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

where $f_i : S \to \mathbb{R}$ 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 $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ 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 $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$, $\boldsymbol{z} \in Z$, 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 $i = 1, \ldots, k$ and $z_j^1 < z_j^2$ for at least one $j = 1, \ldots, k$. 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 $r = 1,\dots,k$:

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

(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

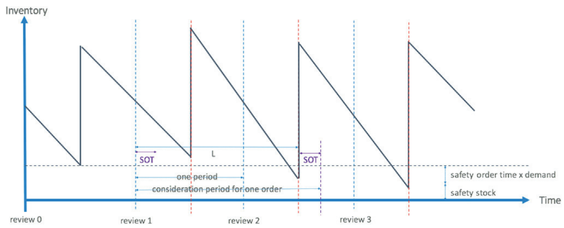


Fig. 1. Illustration of SOT in periodic review policy.

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:

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

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).

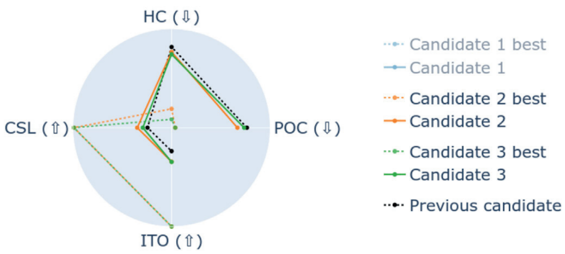


Fig. 3. The spider plot view in the E-NAUTILUS interface. The decision maker is able to select (by clicking on the legend on the right of the plot) one or multiple candidates to be shown simultaneously for comparison. Candidates 2 and 3 have been selected for comparisons in the figure. The best reachable 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.

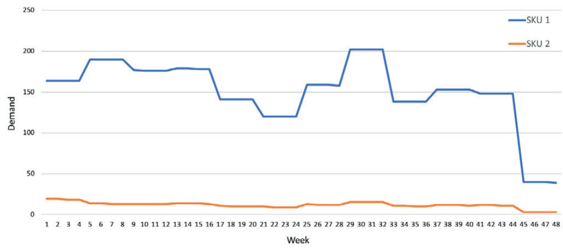


Fig. 4. Demands for SKU 1 (top line) and SKU 2 (bottom line).

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:

$$SOT + \frac{SS}{\mu} \leq MS, \quad (15)$$

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 $z(1) = z(1, 1)$ = (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 $z(2) = z(2, 2)$ = (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 $z(3) = z(3, 3)$ = (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 $z(4) = z(4, 2)$ = (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 $z(5) = z(5, 3)$ = (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 $z(6) = z(6, 3)$ = (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 $z(7) = z(7, 3)$ = (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 $z(8) = z(8, 2)$ = (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 $z(9, 2)$ which had the best CSL value. Then, we set $z(9) = z(9, 2)$ = (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 $z(10) = z(10, 4)$ = (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.

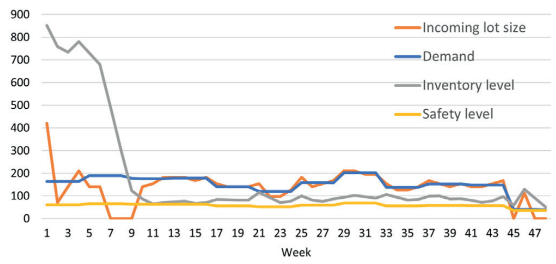


Fig. 5. Result for SKU 1.

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.

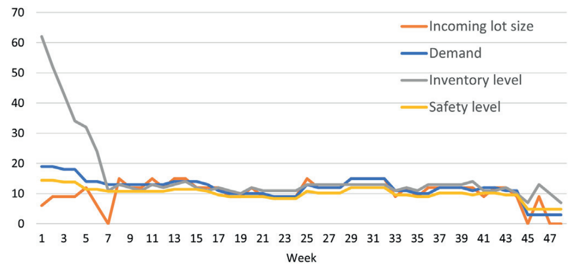


Fig. 6. Result for SKU 2.

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.

PII

INTERACTIVE MULTIOBJECTIVE OPTIMIZATION IN LOT SIZING WITH SAFETY STOCK AND SAFETY LEAD TIME

by

Adhe Kania, Juha Sipilä, Bekir Afsar, and Kaisa Miettinen 2021

In: Mes, M., Lalla-Ruiz, E., Voß, S. (eds),
Computational Logistics. 12th International Conference, ICCL 2021, Proceedings,
pp. 208–221

Interactive multiobjective optimization in lot sizing with safety stock and safety lead time

Adhe Kania[1,3][0000-0001-9586-6226], Juha Sipilä[2][0000-0003-3146-9656], Bekir Afsar[1][0000-0003-3643-2342], and Kaisa Miettinen[1][0000-0003-1013-4689]

[1] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland;
adhe.a.kania@student.jyu.fi
[2] JAMK University of Applied Sciences, School of Technology, Jyväskylä, Finland;
[3] Institut Teknologi Bandung, Faculty of Mathematics and Natural Sciences, Jl Ganesha 10 Bandung 40132, Indonesia

Abstract. In this paper, we integrate a lot sizing problem with the problem of determining optimal values of safety stock and safety lead time. We propose a probability of product availability formula to assess the quality of safety lead time and a multiobjective optimization model as an integrated lot sizing problem. In the proposed model, we optimize six objectives simultaneously: minimizing purchasing cost, ordering cost, holding cost and, at the same time, maximizing cycle service level, probability of product availability and inventory turnover. To present the applicability of the proposed model, we consider a real case study with data from a manufacturing company and apply the interactive NAUTILUS Navigator method to support the decision maker from the company to find his most preferred solution. In this way, we demonstrate how the decision maker navigates without having to trade-off among the conflicting objectives and could find a solution that reflects his preference well.

Keywords: Inventory management · Uncertain demand · Uncertain lead time · Interactive decision making · NAUTILUS Navigator.

1 Introduction

Lot sizing has emerged as one of the key factors for the effective supply chain management. The purpose of lot sizing is to determine an optimal order quantity that minimizes costs while satisfying demand. After Harris's economic order quantity concept for solving a simple lot sizing problem [10], there has been a dramatic increase in interest over the last century in developing lot sizing models to adapt to more complex situations [1,8].

Uncertainties complicate lot sizing problems. In fact, predicting the exact demand for future needs is challenging. Commonly, many companies hold a certain amount of stock, known as a safety stock (SS), as a buffer to cope when demand exceeds the prediction [31]. Another source of uncertainty is the delivery lead time [24]. Companies usually have an agreement with their suppliers for the delivery time, but for many reasons, there can be delays. To overcome this issue, an additional time period, known as a safety lead time (SLT), is defined [31]. During the SLT period, companies keep their stocks available to satisfy the demand.

The problem of determining an optimal SS value has been studied by many researchers [9]. Various methods have been developed [26] to find an optimal value of SS that should be small enough to reduce costs while satisfying demand and guaranteeing a high service level. Most studies expand the cycle service level (CSL) formula [23] to adapt to various conditions. When lead time is uncertain, the CSL formula takes into account the average and standard deviation of the lead time [28]. On the other hand, the problem of finding an optimal SLT value is not as popular as the previous one [7]. In [12], inventory costs are minimized subject to a service level constraint to find an optimal SLT, and an optimization model based on Markov Chain is proposed in [6]. However, there is a lack of formula to control the quality of SLT.

The relationship between lot sizing problems with SS and SLT has been studied in [22]. Keeping stock for SS and SLT increases order quantity, which also increases the costs. Some researchers have studied lot sizing problems with uncertainty on demand and lead time [7]. However, they mostly use statistical tools to handle uncertainty in lot sizing models, but not simultaneously find SS or SLT. Some of them use simulation to find an optimal SS and SLT. There is a lack of integration of a lot sizing problem and problems of determining SS and SLT values in the literature. The problem of integrating lot sizing and SS determination is proposed in [18], but they consider SLT as the input value.

Lot sizing problems naturally include a conflict between minimizing costs and satisfying demand simultaneously. Additional problems of determining SS and SLT increase the conflict because holding more stock for SS and SLT makes the costs higher. For this reason, multiobjective optimization [19] is a good tool to solve lot sizing problems [2]. A multiobjective optimization problem has several mathematically incomparable solutions, called Pareto optimal solutions. Solving a multiobjective optimization problem can be understood as finding the most preferred solution for a decision maker (DM), who has expertise in the problem domain. Interactive methods [20] are regarded as promising because the solution process is iterative and they allow the DM to gain insight into the problem and change his/her preferences during the solution process, thanks to learning. So far, however, there have been a few studies applying interactive multiobjective optimization in lot sizing problems [29].

In this research, we consider a single item multi period lot sizing problem with uncertainty on demand and lead time. The main contributions of this paper are threefold. First, we propose a novel formula, named probability of product availability (PPA), for measuring the quality of SLT to handle unpredicted lead time. Second, we develop a multiobjective optimization model that determines the optimal lot sizes for each period and simultaneously finds the optimal values of SS and SLT. Last but not least, we support a DM to find the most preferred solution for the optimization model by applying an interactive NAUTILUS Navigator method [25].

The proposed multiobjective optimization model has six objectives to optimize simultaneously. Three of them are minimizing cost functions, i.e. purchasing cost (PC), ordering cost (OC), and holding cost (HC). We consider them separately to see trade-offs between objectives. The CSL is maximized to improve safety against demand uncertainty. We propose a PPA formula to assess the quality of SLT to buffer lead time uncertainty, which is maximized in the model. Lastly, inventory turnover (ITO) as the primary performance indicator for inventory management [27] is maximized to measure the effectiveness of this model in handling the inventory system.

Most lot sizing problems are difficult to solve because of their complexity [14]. In this paper, we use the interactive NAUTILUS Navigator method [25]. The strength of this method in handling computationally expensive problems meets the need of this kind of problem. Another strength of this method is allowing the DM to find his/her most preferred solution without sacrifices, which meets the needs of the DM. In this, the strategy is starting from a bad point and improving all objectives simultaneously. We use real data from a manufacturing company and a real DM to prove the validity of our proposed model. Finally, we support the DM to find the most satisfying solution for him by using this method.

The remainder of the paper is organized as follows. Section 2 reviews the basic concepts of multiobjective optimization and the NAUTILUS Navigator method. Then, the proposed multiobjective optimization model is presented in Section 3. In Section 4, the case study together with the real data from a manufacturing company is described, following by results and analysis of the decision making process using NAUTILUS Navigator. Finally, conclusions and discussions of possible extensions are presented.

2 Background on Multiobjective Optimization

In this section, we briefly review the basic concepts and definitions related to multiobjective optimization, followed by the NAUTILUS Navigator method.

2.1 Basic Concepts and Definitions

A multiobjective optimization problem can be formulated in the following form:

$$\begin{array}{ll} \text{minimize} & \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_k(\boldsymbol{x}))^T \\ \text{subject to} & \boldsymbol{x} \in S, \end{array} \tag{1}$$

where $k \geq 2$ objective functions, $f_i : S \to \mathbb{R}$ for $1 \leq i \leq k$, are simultaneously optimized. The vector of decision variables $\boldsymbol{x} = (x_1, ..., x_n)^T$ belongs to the feasible region $S \subset \mathbb{R}^n$, which is formed by constraints. The image of the feasible region $Z = \boldsymbol{f}(S), Z \subset \mathbb{R}^k$ is called a feasible objective region, which is formed by the vectors of objective values $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_k(\boldsymbol{x}))^T$, $\boldsymbol{z} \in Z$, $\boldsymbol{x} \in S$.

Because of the conflicting objectives, a multiobjective optimization problem (1) has several different solutions, called Pareto optimal solutions, which reflect the trade-offs among the conflicting objectives. A solution $\boldsymbol{z}^1 \in Z$ is said to dominate another solution $\boldsymbol{z}^2 \in Z$ if $z_i^1 \leq z_i^2$ for all $i = 1, ..., k$ and $z_j^1 < z_j^2$ for at least one $j = 1, ..., k$. A solution $\boldsymbol{z} \in Z$ is called a Pareto optimal solution if $\boldsymbol{z}$ is not dominated by any other solution. The lower and upper bounds of the Pareto optimal solutions are called an ideal point $\boldsymbol{z}^*$ and a nadir point $\boldsymbol{z}^{nad}$, respectively, which reflect the best and the worst values that each objective function in the Pareto optimal solutions can achieve.

Pareto optimal solutions are incomparable mathematically. Additional preference information from a DM is needed to identify the most preferred solution as the final solution. A DM is an expert who has a responsibility to make a decision in the problem domain, who is usually a supply chain manager in lot sizing. The preference information from the DM can be incorporated before the optimization process (a priori methods), after having generated a representative set of Pareto optimal solutions (a posteriori methods), or during an iterative optimization process (interactive methods) [19]. The advantages of interactive methods, which allow the DM to learn different aspects of the problem during the solution process and change their preferences during the solution process if desired, are the main reasons we chose this type of methods. Many interactive methods have been developed [20]. In this paper, we apply the NAUTILUS Navigator method [25] because of its ability in handling computationally expensive problems and the possibility to find the most preferred solution without trading-off. This is important since DMs sometimes get anchored around the initial solution and a trade-off free method avoids anchoring.

2.2 NAUTILUS Navigator

The NAUTILUS Navigator method combines the idea of NAUTILUS methods [21] to avoid trading-off and navigation ideas elaborated in [11]. Due to the fact that people do not respond similarly to losses and gains [15], trading-off among Pareto optimal solutions causes some decisional stress to the DM [17]. Motivated by this fact, NAUTILUS methods start from the worst possible objective function values and iteratively gain in all objectives without sacrificing any of the current values. Methods in the NAUTILUS family [21] differ in the way used to interact with the DM to find the final solution. NAUTILUS Navigator uses navigation to direct the movement from the worst starting point, which is the nadir point or any undesirable point provided by the DM, to a Pareto optimal solution as the final solution. In this process, the DM specifies a desirable value for each objective function, which are the components of a reference point, as a search direction to direct the movement towards desired Pareto optimal solutions. During the navigation process, the DM can change the reference point, the movement speed, or even go backwards if he/she wishes so.

To handle computationally expensive problems, a set of Pareto optimal solutions is generated before the interactive process starts. The generation may take time because of expensive functions, but it is done without involving the

DM. Any a posteriori methods can be used to generate a set of Pareto optimal solutions or a set that approximates Pareto optimal solutions. When involving the DM, the navigation process takes place using this set without solving the original computationally expensive problem. This allows showing real-time movement without waiting times to the DM. The detailed algorithm can be seen in [25].

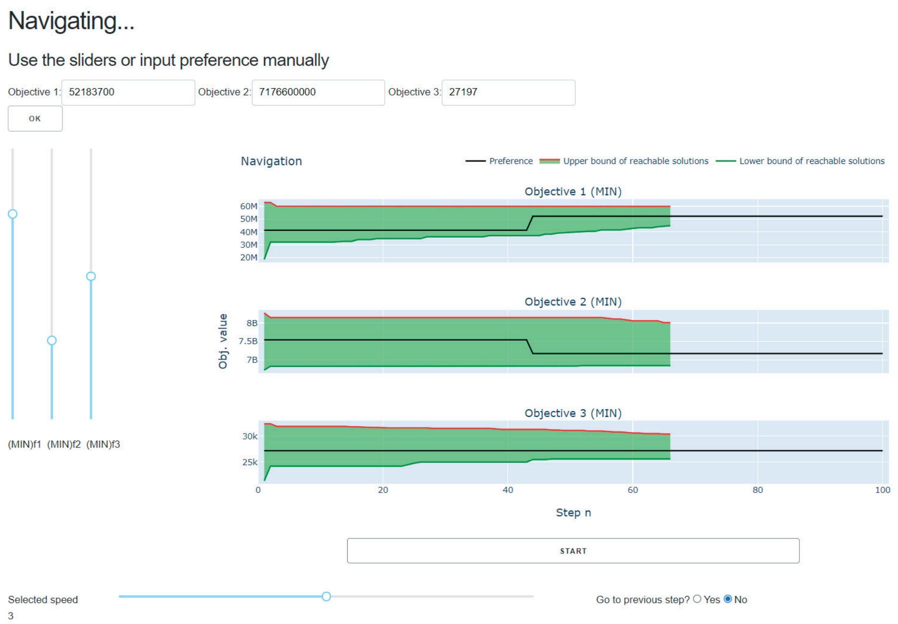


Fig. 1. GUI of the NAUTILUS Navigator method

A graphical user interface (GUI) is important for NAUTILUS Navigator to visualize the navigation process. Figure 1 shows the available GUI that can be freely downloaded from `https://desdeo.it.jyu.fi`. The DM provides his/her preferences using the sliders on the left side or inputs values in text boxes at the top. The green area in the graph shows the reachable ranges, which are the best and the worst objective function values, that each objective can reach from the current step without sacrifices in any objectives. Thus, the reachable ranges shrink when approaching Pareto optimal solutions. Whenever the DM wants to change his/her preference, he/she can stop the process and change the reference point. The black lines in the middle of the graphs show the positions of the components of the reference point. The DM is allowed to jump to any previous step using the radio button in the bottom right. He/she then needs to provide which step to go to and re-specify his/her preferences in order to define a new direction. The DM can navigate until he/she finds his/her most preferred

Pareto optimal solution at the end of the solution process. In that case, the ranges shrink to a single point.

3 Problem Formulation

We study a lot sizing problem for a single item with a single supplier and in multiple time periods. We follow a periodic review policy, where orders are reviewed over discrete time periods $t = 1, ..., T$. The order quantity ($Q(t)$) is reviewed at the beginning of period t, and the order arrives after a stochastic lead time. The following assumptions are made throughout this paper.

1. The predicted demand during period t ($D(t)$) follows a normal distribution with a mean μ and a standard deviation σ. The demand in each period is independent of other periods.
2. The lead time follows a normal distribution with a mean L and a standard deviation s.
3. The price for purchasing one unit of item (p) is constant in all time periods and does not depend on the order quantity.
4. The cost for a single order is c without any capacity limit.
5. The cost of holding one unit of item (h) is constant throughout all time periods.
6. There is no backorder cost involved.
7. There is an agreement between the company and the supplier that the company must order with a minimum order quantity moq and it rounds up by a rounding value r. Therefore, the order can only be placed by following the formula $moq + a\,r$ for any integer $a \geq 0$.

3.1 Safety Stock and Safety Lead Time Formulation

As said, we focus on the lot sizing problem with uncertainty in demand and lead time. Many researchers have utilized a SS to protect against demand uncertainty and a SLT to handle lead time uncertainty [16]. A SS means keeping more stocks as a buffer against demand fluctuations. To control the amount of SS, the cycle service level (CSL) formula is applied [4]. CSL is the probability of not hitting a stockout in a replenishment time (RT). A RT is a time needed to refill the stock, that is from the arrival of one order to the arrival of the next one. We set $RT = 1+SLT$ since we order in each period and prepare for late delivery in the SLT period. To prevent stockout during a RT, the difference between an actual demand (D^*_{RT}) and a predicted demand (D_{RT}) must be less than SS. We adopt the CSL formula for demand and lead time uncertainty [28] with our definition of RT, which can be written as follows:

$$\begin{aligned} CSL &= P(D^*_{RT} \leq D_{RT} + SS) \\ &= F(D_{RT} + SS, D_{RT}, \sigma_{RT}) = F\left(\frac{SS}{\sigma_{RT}}\right), \end{aligned} \tag{2}$$

where F is the standard normal distribution function and σ_{RT} is the standard deviation of demand during a RT, which can be formulated as $\sigma_{RT} = \sqrt{\sigma^2(1+SLT)+\mu^2 s^2}$.

A SLT is assigned to handle unpredicted lead time. During the SLT period, the availability of stock to cover predicted and unpredicted demand must be guaranteed. Therefore, we consider an additional SLT period in the fill rate (FR) constraint to secure the availability of the stock during SLT to cover the predicted demand. A SLT period is also considered in CSL to buffer unpredicted demand during SLT. However, if the order arrives after the SLT period, the stockout may occur. Therefore, it is important to decide an optimal SLT value with a low possibility of having stockout. In this paper, we propose the probability of product availability (PPA) formula to measure the quality of SLT. PPA is defined as the probability of not having stockout because of the late delivery, which occurs when the actual order arrives during the period $L+SLT$. The PPA formula can be written as follows:

$$\begin{aligned} PPA &= P(\text{actual delivery time} \leq L+SLT) \\ &= F(L+SLT, L, s) = F\left(\frac{SLT}{s}\right). \end{aligned} \tag{3}$$

This formula can be used to find the SLT value by defining an appropriate PPA level.

3.2 Multiobjective Optimization Model

As said, we propose a multiobjective optimization model with six objectives, three to minimize and three to maximize. The main goal of this model is to find the order quantity of each period $(Q(t), t=1,...,t_n)$ together with SS and SLT values with the best balance between the objective functions. We define $I(t)$ as the inventory level at the end of period t where $I(t) = I(t-1)+Q(t-\lfloor L \rfloor)-D(t)$, and $Y(t)$ as the order indicator where $Y(t)=1$ if the order is placed $(Q(t)>0)$, otherwise $Y(t)=0$. The proposed optimization model can be written as follows.

$$\text{min} \quad PC = \sum_t Q(t)\, p,\ OC = \sum_t Y(t)\, c,\ HC = \sum_t \frac{I(t-1)+I(t)}{2}\, h,$$

$$\text{max} \quad CSL\ (2),\ PPA\ (3),\ ITO = \sum_t \frac{D(t)}{(I(t-1)+I(t))/2},$$

$$\text{s.t.} \quad \frac{I(t-1)+\sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{\sum_{j=t}^{t+\lfloor P \rfloor} D(j) + (P-\lfloor P \rfloor) D(\lceil P \rceil)} \geq 1\ \text{, for } t=1,...,T, \tag{4}$$

$$Q(t) = Y(t)\,(moq + a\,r)\ \text{, for any integer } a \geq 0 \text{ and } t=1,...,T, \tag{5}$$

$$SS \geq 0 \text{ and } SOT \geq 0. \tag{6}$$

Following the dynamic lot sizing problem [14,30], three types of cost are considered: PC, OC, and HC. We consider them separately to see the trade-offs. Minimizing PC implies minimizing HC, but OC has a trade-off with HC because ordering the same amounts of items many times makes OC higher and HC lower. For inventory management purposes, it is important to understand both HC and OC. In order to prevent partial optimization, which could be the case if only total costs were measured, it is important to separate them. When targeting at low HC only, one can be misled, as then there could be a temptation to order more often, resulting in higher OC.

We maximize CSL to prevent stockout because of the demand uncertainty and maximize PPA to avoid stockout due to late delivery. Keeping a high value of SS raises the CSL but PC and HC increase, which is a conflict as we need to maximize CSL but minimize OC and HC. Having a long SLT increases the PPA but decreases CSL with the same SS value. Then PPA has a trade-off with CSL, PC and HC. Maximizing ITO is our last objective function. To have a high ITO, the order must be as close to the demand as possible in order to hold less stock, which has a trade-off with OC. Furthermore, ITO has a trade-off with CSL and PPA as less stock is needed to have a high ITO, but CSL and PPA need more stock to have better safety in handling uncertainties.

FR represents customer service for an inventory control system. It is defined as the fraction of orders that are filled from stock [13]. It is an important indicator in daily operations. In the proposed model, FR is the first constraint (4) to fulfill the predicted demand. In each period, we guarantee that our stock (excluding SS) can satisfy the predicted demand. The consideration period for one order (P) in the periodic review policy is $1 + L$ [4], but an additional SLT period is also considered to ensure the stock availability during SLT. Thus, we set $P = 1 + L + SLT$. FR is a fraction between available stock without SS and the predicted demand during P. When FR is at least one, the stock availability to handle the predicted demand is guaranteed. Furthermore, we ensure that all orders follow the agreement of minimum order quantity and rounding value in constraint (5), while constraint (6) is defined to confine the lower bounds of SS and SLT.

4 Computational Results

We consider a case study from a manufacturing company to demonstrate the applicability of the proposed model. We apply the interactive NAUTILUS Navigator method to support the supply chain manager of the said company, acting as the DM, to find his most preferred solution without trading-off.

4.1 Information about the Case

We review a weekly single item lot sizing problem for 41 weeks. Thus, the optimization model has 43 integer decision variables, including weekly order quantities, SS and SLT. We received data of an item, which is a component of the company's product. The data is generated from the company's planning system. The data contains current inventory information for the item as well as a consumption projection according to the company's production plan. Based on the data, the price to purchase one unit of the item is €91.18, the cost for a single order is €200, and the cost of holding one unit of item is ten percent of the price annually. The lead time for this item is 6 weeks, with a standard deviation $s = 0.93$ days. The company has made a prediction for the weekly demand data based on its historical data, which varies with a mean $\mu = 116.22$ and a standard deviation $\sigma = 29.04$. The opening inventory is 312 units and the company has made previous orders for the next six weeks, which are $(48, 119, 120, 120, 48, 96)$. Based on the agreement between the company and the supplier, the company must place an order with a minimum of 48 units and round by 48 units.

As a request from the DM, bounds for SS and SLT were defined as additional constraints. The DM was only interested in SS values lower than μ and SLT values below four days. He also requested to see at least one day SLT or one day's worth of demand for SS, which is $\mu/5$. Furthermore, low ITO values below ten were not interesting for the DM.

As said, a GUI plays an important role in NAUTILUS Navigator. A few modifications of the available GUI were done in this research to make the GUI more useful for the DM in this case. The DM preferred to see the probability of product unavailability (PPU) rather than PPA. Thus, we switched to minimize $PPU = 1 - PPA$ in the fifth objective. Furthermore, the DM wanted to see the information of days of stock (DoS). DoS is an inventory performance indicator describing the number of days needed to sell an item. DoS is calculated as the number of days in one year (we use 254 working days) divided by ITO.

4.2 Computational results

As described in Section 2.2, the starting point of the NAUTILUS Navigator method is a set of pre-generated solutions. As said, lot sizing problems are computationally expensive problems. Because of their complexity, many researchers use metaheuristic methods, like evolutionary algorithms, to solve various problems of lot sizing [14]. In this paper, we applied NSGA-III [5] by using the pymoo framework [3] because of its ability to solve constrained multiobjective optimization problems with integer variables. Evolutionary algorithms cannot guarantee Pareto optimality but can generate sets of solutions where no solution dominates the others.

Some strategies were needed to generate a large amount of nondominated solutions. Because a single run of NSGA-III was not able to generate enough solutions, we ran the algorithm several times with different initial populations. Furthermore, to get more solutions, various parameters of evolutionary operators were used that were available in the framework. Finally, all solutions were combined, dominated solutions were deleted, and 1503 nondominated solutions were obtained that approximate Pareto optimal solutions.

The DM started the navigation process by investigating the reachable ranges for the first step, which were represented by the ideal point and the nadir point initially derived from the set. With the bounds defined by the DM, the ideal point was $\boldsymbol{z}^* = (358\ 884.48,\ 1\ 000,\ 674.73,\ 0.9945,\ 0,\ 97.45)$ and the nadir point was $\boldsymbol{z}^{nad} = (367\ 637.76,\ 6\ 800,\ 4\ 782.04,\ 0.5,\ 0.5,\ 10.19)$ (remember that the fourth and sixth objectives are to be maximized and the others are to be minimized). Initially, the DM wanted to set the ideal point as the reference point to investigate how the navigation ran and which Pareto optimal solutions can be found if he wanted all the objectives to navigate towards their best values.

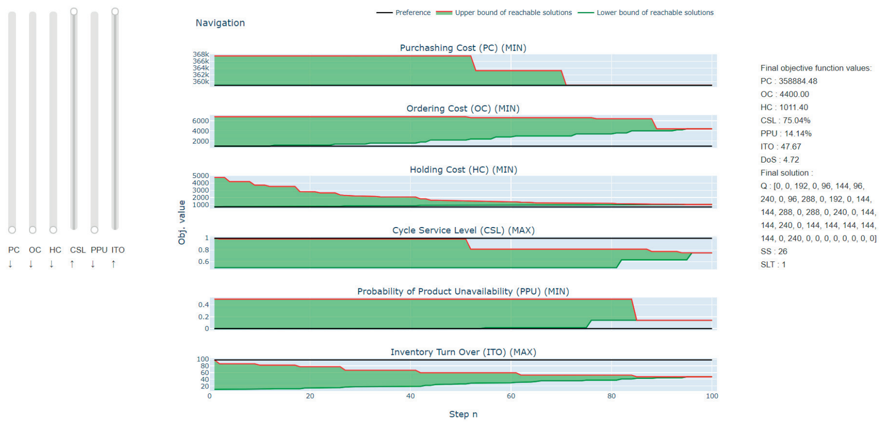


Fig. 2. A Pareto optimal solution for the ideal point as the reference point

Because of the trade-offs among the objectives, getting the best possible values for all objectives is naturally impossible, but, the DM navigated till the Pareto optimal solution $\boldsymbol{z} = (358\ 884.48,\ 4\ 400,\ 1\ 011.40,\ 0.7504,\ 0.1414,\ 47.67)$ was reached. Thus, the reachable range was finally a single point. Figure 2 shows this navigation. The DM analyzed that, in step 52, there was a significant decrease of the upper bound for the reachable CSL values to 0.8116, and the ITO reachable range shrunk with the upper bound 59.53. Because of this, the DM decided to go backwards to step 50 and provided new preferences.

The DM wanted to keep the ITO in the best value at this step, which was 59.53. He then set the components of the reference point for PC and OC to their worst values, and keep the other components as their best reachable values at this step. Therefore, the new reference point was (367 637.76, 6 800, 901.98, 0.9835, 0, 59.53). He let the navigation continue until the end to check the Pareto optimal solution that could be reached. The Pareto optimal solution obtained was $\boldsymbol{z} = (363\ 261.12,\ 6\ 000,\ 1\ 108.19,\ 0.8437,\ 0.0159,\ 39.25)$. He found the CSL value better but it was not satisfactory enough for him. He learned that the upper bound of the CSL's reachable values started to decrease at step 80. He then decided to return to this step to set a new reference point.

The DM navigated with different desired values of ITO to observe how much he needed to sacrifice in ITO to get better values for CSL. He returned to step 80 a few times with different desired values for ITO, but he only got 0.9041 as the best value for CSL. He decided to go further backwards to step 16 because the upper bound of ITO and HC in reachable values had a significant decrease after this step. He set all cost objectives in their worst reachable values, CSL and PPU in their best reachable values, and ITO=48. He let the reachable ranges shrink till the Pareto optimal solution $\boldsymbol{z} = (363\ 261.12,\ 6\ 400,\ 1\ 183.94,\ 0.9366,\ 0.0159,\ 35.68)$. The DM found that the CSL value was not satisfactory enough.

The DM realized that CSL had a trade-off with PPU, and he needed to relax PPU to get better CSL. He decided to return to step 75 when the CSL decreased. He then relaxed the ITO value to the worst reachable value, and got the Pareto optimal solution $\boldsymbol{z} = (363\ 261.12,\ 5\ 800,\ 1\ 066.10,\ 0.9272,\ 0.1414,\ 42.69)$. He was happy with the improvement of ITO but was still curious to find a better CSL value.

The DM wanted to investigate how much he needed to sacrifice in ITO when he desired to improve CSL. He then decided to go to the very first step and set his preferences at the best reachable value for CSL and the worst reachable values for costs and PPU. For ITO, he set 40 as the desired level. He let the navigation converge to a single solution. He got the best CSL value and the Pareto optimal solution was $\boldsymbol{z} = (367\ 637.76,\ 5\ 800,\ 1\ 061.90,\ 0.9945,\ 0.5,\ 42.94)$. He was very happy with this solution. He thought that the CSL value was very good and the other objective values were acceptable. He decided to accept this solution as the final one.

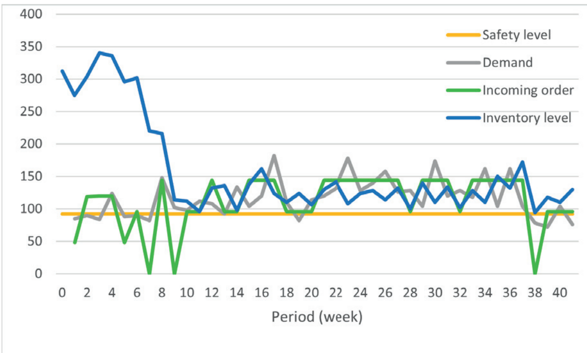


Fig. 3. The decision variables corresponding to the final solution

The decision variables corresponding to the final solution for order quantities can be seen in Figure 3. The other decision variables were $SS = 92$ and $SLT = 0$. The green line in Figure 3 shows the incoming order quantities for each week, which are the previously set order data for $t = 1, ..., 6$ and the optimized order quantities $Q(t-L)$ for $t = 7, ..., 41$. The inventory level in the blue line shows that during the first six weeks, which cannot be controlled by the model due to the lead time, the company had excess inventory. The inventory level then decreased and followed the demand quantity to have a higher ITO, which is a useful indicator for inventory management and planning purposes.

By deepening his understanding of the interdependencies between conflicting objectives, the DM learned a lot from his own area of responsibility as a supply chain manager and also gained the confidence to modify his original preferences. At first, he was not willing to sacrifice on any objectives, but during the decision making process, there was a growing awareness that not everything can be achieved, but sacrifices have to be made. These included, among other things, the CSL and ITO. However, in his day-to-day operations, ITO is a goal set by the company's top management. Therefore, deviating from this objective must be strongly justified to the management.

As a result of the learning process, the DM gained confidence in setting his preferences, and thus multiobjective optimization and NAUTILUS Navigator supported his understanding and ability to justify his decisions. The DM greatly appreciated the fact that as the decision making process progressed, he constantly saw the navigator's results and understanding of achieving objectives, which guided him in setting his preferences. The possibility to stop the process at any time and the feature to go backwards in the navigator, were, in his view, excellent opportunities to make decisions easily. The GUI of the navigation and the real-time updating of the results also supported his decision making. The navigator graphs and the sliders for setting the reference point were, in the DM's view, a clear advantage in support of decision making. The whole process was so instructive and professionally useful.

As can be seen in Figure 3, the inventory level was significantly reduced from its original level. The DM commented that this is a typical example of decisions being made in the past "for the sake of certainty", where typically stock levels tend to rise. NAUTILUS Navigator as a method responded precisely to the need for decisions to be based on calculations rather than assumptions. The DM was pleased with the result of the objective function values, as well as the corresponding decision variables. Overall, the DM was satisfied with the results and operation of NAUTILUS Navigator and found an interactive method very suitable for learning. He is willing to adopt the method more widely for inventory planning and control, especially for critical items.

5 Conclusions

In this paper, we considered a single item multi period lot sizing problem in a periodic review policy under a stochastic environment on demand and lead time. We used a SS to handle uncertainty on demand and CSL to measure the quality of SS. To handle uncertainty on lead time, a SLT was used and we proposed the PPA formula to measure the quality of SLT. The aim of this paper was to integrate the lot sizing problem with the problem of determining the optimal values of SS and SLT. We developed a multiobjective optimization model to solve the integrated lot sizing problem. Six objectives were optimized simultaneously to find the optimal order quantity in each period and at the same time determine the optimal values of SS and SLT.

Real data from a manufacturing company was used to demonstrate the applicability and usefulness of the proposed model. A supply chain manager from the said company acted as the DM to draw managerial insights into the decision making process. The interactive NAUTILUS Navigator method was successfully applied to solve our integrated computationally expensive lot sizing problem. The DM appreciated the navigation process that allowed him to learn during the decision making process and find the most satisfying solution for him. He confirmed the validity of the solution and found it useful for his daily operation.

For future research, considering many items would present more computational challenges but meet the needs of real industrial problems. A company may have thousands of items that are impossible to consider separately. Another possible future research topic is to address the variation of price based on the order quantity, or integrating the model with the problem of determining minimum order quantity and rounding value.

Acknowledgements 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 (grants 322221 and 311877). 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.

References

1. Andriolo, A., Battini, D., Grubbström, R.W., Persona, A., Sgarbossa, F.: A century of evolution from Harris's basic lot size model: Survey and research agenda. International Journal of Production Economics **155**, 16–38 (2014)
2. Aslam, T., Amos, H.C.N.: 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 (2010)
3. Blank, J., Deb, K.: Pymoo: Multi-objective optimization in Python. IEEE Access **8**, 89497–89509 (2020)
4. Chopra, S., Meindl, P.: Supply chain management: strategy, planning, and operation. Pearson, 6 edn. (2016)
5. Deb, K., Jain, H.: 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 (2014)
6. Dolgui, A., Ould-Louly, M.A.: A model for supply planning under lead time uncertainty. International Journal of Production Economics **78**(2), 145–152 (2002)
7. Dolgui, A., Prodhon, C.: Supply planning under uncertainties in MRP environments: A state of the art. Annual Reviews in Control **31**(2), 269–279 (2007)
8. Glock, C.H., Grosse, E.H., Ries, J.M.: The lot sizing problem: A tertiary study. International Journal of Production Economics **155**, 39–51 (2014)

9. Gonçalves, J.N., Sameiro Carvalho, M., Cortez, P.: Operations research models and methods for safety stock determination: A review. Operations Research Perspectives **7**, 100164 (2020)
10. Harris, F.W.: How many parts to make at once. Factory, The Magazine of Management **10**, 135–136 (1913)
11. Hartikainen, M., Miettinen, K., Klamroth, K.: Interactive Nonconvex Pareto Navigator for multiobjective optimization. European Journal of Operational Research **275**(1), 238–251 (2019)
12. Hegedus, M.G., Hopp, W.J.: Setting procurement safety lead-times for assembly systems. International Journal of Production Research **39**(15), 3459–3478 (2001)
13. Hopp, W.J., Spearman, M.L.: Factory Physics. Waveland Press, Inc, 3 edn. (2008)
14. Jans, R., Degraeve, Z.: Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. European Journal of Operational Research **177**(3), 1855–1875 (2007)
15. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. Econometrica **47**(2), 263–291 (1979)
16. van Kampen, T.J., van Donk, D.P., van der Zee, D.J.: Safety stock or safety lead time: coping with unreliability in demand and supply. International Journal of Production Research **48**(24), 7463–7481 (2010)
17. Korhonen, P., Wallenius, J.: Behavioural issues in MCDM: Neglected research questions. Journal of Multi-Criteria Decision Analysis **5**(3), 178–182 (1996)
18. Kumar, K., Aouam, T.: Integrated lot sizing and safety stock placement in a network of production facilities. International Journal of Production Economics **195**, 74–95 (2018)
19. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
20. Miettinen, K., Hakanen, J., Podkopaev, D.: Interactive nonlinear multiobjective optimization methods. In: Greco, S., Ehrgott, M., Figueira, J. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 931–980. Springer, 2 edn. (2016)
21. Miettinen, K., Ruiz, F.: NAUTILUS framework : Towards trade-off-free interaction in multiobjective optimization. Journal of Business Economics **86**(1), 5–21 (2016)
22. Molinder, A.: Joint optimization of lot-sizes, safety stocks and safety lead times in an MRP system. International Journal of Production Research **35**(4), 983–994 (1997)
23. New, C.: Safety stocks for requirements planning. Production and Inventory Management **12**, 1–18 (1975)
24. Pahl, J., Voß, S., Woodruff, D.L.: Production planning with load dependent lead times: an update of research. Annals of Operations Research **153**, 297–345 (2007)
25. Ruiz, A.B., Ruiz, F., Miettinen, K., Delgado-Antequera, L., Ojalehto, V.: NAUTILUS Navigator: free search interactive multiobjective optimization without trading-off. Journal of Global Optimization **74**, 213–231 (2019)
26. Schmidt, M., Hartmann, W., Nyhuis, P.: Simulation based comparison of safety-stock calculation methods. CIRP Annals **61**(1), 403–406 (2012)
27. Silver, E.A., Pyke, D.F., Thomas, D.J.: Inventory and production management in supply chains. CRC Press, 4 edn. (2017)
28. Talluri, S., Cetin, K., Gardner, A.J.: Integrating demand and supply variability into safety stock evaluations. International Journal of Physical Distribution and Logistics Management **34**(1), 62–69 (2004)

29. Torabi, S., Hassini, E.: An interactive possibilistic programming approach for multiple objective supply chain master planning. Fuzzy Sets and Systems **159**(2), 193–214 (2008)
30. Wagner, H.M., Whitin, T.M.: Dynamic version of the economic lot size model. Management Science **5**, 89–96 (1958)
31. Whybark, D.C., Williams, J.G.: Material requirements planning under uncertainty. Decision Science **7**(4), 595–606 (1976)

PIII

DETERMINING MINIMUM ORDER QUANTITY AND LOT SIZES FROM THE BUYER'S PERSPECTIVE WITH INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

by

Adhe Kania, Bekir Afsar, Kaisa Miettinen, and Juha Sipilä

Submitted to a journal

Request a copy from the author.

PIV

DESMILS: A DECISION SUPPORT APPROACH FOR MULTI-ITEM LOT SIZING USING INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

by

Adhe Kania, Bekir Afsar, Kaisa Miettinen, Juha Sipilä 2023

Journal of Intelligent Manufacturing (to appear),
https://doi.org/10.1007/s10845-023-02112-5

Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-023-02112-5

DESMILS: a decision support approach for multi-item lot sizing using interactive multiobjective optimization

Adhe Kania[1,3] · Bekir Afsar[1] · Kaisa Miettinen[1] · Juha Sipilä[2]

Received: 6 April 2022 / Accepted: 8 March 2023

Abstract

We propose a decision support approach, called DESMILS, to solve multi-item lot sizing problems with a large number of items by using single-item multiobjective lot sizing models. This approach for making lot sizing decisions considers multiple conflicting objective functions and incorporates a decision maker's preferences to find the most preferred Pareto optimal solutions. DESMILS applies clustering, and items in one cluster are treated utilizing preferences that the decision maker has provided for a representative item of the cluster. Thus, the decision maker provides preferences to solve the single-item lot sizing problem for few items only and not for every item. The lot sizes are obtained by solving a multiobjective optimization problem with an interactive method, which iteratively incorporates preference information and supports the decision maker in learning about the trade-offs involved. As a proof of concept to demonstrate the behavior of DESMILS, we solve a multi-item lot sizing problem of a manufacturing company utilizing their real data. We describe how the supply chain manager as the decision maker found Pareto optimal lot sizes for 94 items by solving the single-item multiobjective lot sizing problem for only ten representative items. He found the solutions acceptable and the solution process convenient saving a significant amount of his time.

Keywords Lot sizes · Inventory management · Interactive method · Multiple criteria optimization · NIMBUS

Introduction

In a strategic buyer–supplier relationship, both buyer and supplier aim to create a benefit in order to gain a competitive advantage (Tanskanen & Aminoff, 2015). Lot sizing is central to the cost-effectiveness of inventory management in manufacturing companies and, therefore, it has motivated much research in production planning and control. Beginning with the economic order quantity concept of Harris (1913) in 1913, numerous variants and extensions of lot sizing models have been proposed in the literature [see e.g. the surveys (Andriolo et al., 2014; Glock et al., 2014)]. Integrating a lot sizing problem to other related problems has also been studied, such as integration with scheduling (Copil et al., 2017), supplier selection (Aissaoui et al., 2007), cutting stock problem (Melega et al., 2018), manufacturing and remanufacturing (Naeem et al., 2013), or safety strategy placement (Kania et al., 2022).

Lot sizing problems focus on the trade-off of meeting customer demand while minimizing cost. It naturally introduces conflicting objective functions even though many studies in the literature consider it as a single objective optimization problem and set demand as a constraint. Dealing with more complex situations such as demand and lead time uncertainty or integrating lot sizing problems with other problems introduce more conflicting objective functions. Therefore, some studies consider more than one objective function in their lot sizing problems [see e.g. Aslam Amos (2010), Heikkinen et al. (2021) and Kania et al. (2021)].

Tools that support optimization of multiple (conflicting) objective functions belong to the field of multiobjective optimization (Miettinen, 1999). Because of multiple objective

✉ Adhe Kania
adhe.a.kania@student.jyu.fi

1 University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland

2 JAMK University of Applied Sciences, School of Technology, Jyväskylä, Finland

3 Institut Teknologi Bandung, Faculty of Mathematics and Natural Sciences, Jl Ganesha 10, Bandung 40132, Indonesia

functions to be optimized simultaneously, a multiobjective optimization problem typically does not have one optimal solution, but a set of compromise solutions, called Pareto optimal solutions. A solution is Pareto optimal if none of the objective functions can be improved without impairing at least one of the others. The goal of multiobjective optimization is to support a decision maker (DM), who is an expert in the problem domain, to find his/her most preferred solution among the Pareto optimal solutions. Interactive methods (Miettinen et al., 2016), which iteratively incorporate the DM's preferences, are regarded as promising to find a most preferred solution for the DM. These methods allow the DM to learn about the problem and trade-offs among the objective functions during the decision making process. The DM is also allowed to adjust his/her preferences and improve the solution until he/she finds the most preferred solution for him/her. So far, however, as shown in the survey in Heikkinen et al. (2021), there have been only few studies applying interactive multiobjective optimization in lot sizing problems.

Most studies in lot sizing consider a single item only (Brahimi et al., 2017), but in reality, companies need to decide order quantities for many items, or even thousands of items for a big company. Therefore, some studies focus on multi-item lot sizing problems. However, most of them model their problem as an optimization problem with a single objective function. In Absi et al. (2013), a multi-item capacitated lot sizing problem with setup times and lost sales is studied. The objective function to be optimized in this paper is the total cost that aggregates production, setup, inventory and shortage costs. In Li et al. (2012), a multi-item capacitated dynamic lot sizing problem is considered and a framework proposed to minimize a single objective function representing total costs, including production cost, inventory holding cost and fixed setup cost. A multi-item capacitated lot sizing problem with remanufacturing is dealt with in Cunha et al. (2019). The authors propose a method to solve their mixed integer lot sizing problem to minimize the total production/remanufacturing, setup and holding costs.

Only few researchers used multiobjective optimization to solve their multi-item lot sizing problems. A multi-item capacitated lot sizing problem with setup times, safety stock and demand shortage costs were studied in Mehdizadeh et al. (2016). The authors modeled an optimization problem with two objective functions to minimize total costs and simultaneously minimize required storage space. In Ammar et al. (2020), a multi-item capacitated lot sizing problem with consideration of setup times and backlogging was addressed, and an optimization problem with two objective functions was solved to minimize total costs and total inventory level of items.

To the best of our knowledge, the literature in multi-item lot sizing problems has considered a sum of functions (e.g., total costs) for all items as one objective function (e.g., minimizing total costs). This kind of a model treats each item similarly and cannot accommodate different preferences from the DM in lot sizing decisions for different items. In fact, the DM may have different preferences in his/her lot sizing decision e.g., for items with a low and a high demand or items with a low and a high price. It is demonstrated in Kania et al. (2022) that the DM had different preferences for two items with a high and a low demand. In the case considered, he paid more attention to inventory turnover values for the item with a high demand and a low price, but was more concentrated on cycle service level for the item with a low demand and a high price.

A single decision making process cannot accommodate difference preferences in deciding lot sizing for different items. However, repeating the decision making process for every single item is laborious. In machine learning, clustering divides a set of objects into clusters, such that objects in the same clusters are more similar to each other than objects in the different clusters [see e.g. Xu and Tian (2015) and Xu and Wunsch (2005)]. This clustering idea has inspired us to divide items into clusters, so that one cluster can be considered with similar preference information, and, therefore the decision making process is only conducted once for each cluster. The aim is to decrease the amount of effort required from the DM.

In this paper, we propose an approach, called DESMILS, to support decision making in multi-item multiobjective lot sizing problems. This approach expects the DM to solve a single-item multiobjective lot sizing problem for a small amount of selected items. Then the preferences obtained from the DM are accommodated in deriving lot sizes for the other items. Therefore, the need of repeating a decision making process for each item separately is avoided. DESMILS enables applying interactive multiobjective optimization methods in solving multi-item lot sizing problems. It can also be applied for any variant or extension of single-item lot sizing models (mentioned earlier).

The idea of the novel approach is to cluster items so that items in the same cluster can be treated with similar preferences in the lot sizing decision. Hence, the DM is only required to do the decision making process for one representative item of each cluster, instead of every single item. The DM can choose the number of clusters which implies the number of decision making processes that he/she is convenient to conduct (for the representatives of each cluster). Finally, the preference information from the DM is utilized to find the optimal lot sizes for remaining items.

As a proof of concept, we demonstrate the approach with a real problem in a manufacturing company. The supply chain manager from the company acted as the DM. In the case study, we use the lot sizing problem integrated with safety strategy placement proposed in Kania et al. (2022). We demonstrate that DESMILS could successfully support the DM in finding the most preferred lot sizes for 94 items. The DM appreciated the benefit of DESMILS to find solutions that best represent his preferences without having to conduct 94 decision making processes individually. Instead, he only needed to repeat the decision making process for few times (an acceptable number for him). This saved much time and effort.

For measuring the performance of supply chain management in lot sizing, key performance indicators (KPIs) are widely used (Akyuz & Erkan, 2010). Managerial insight here is that objective functions are as such useful KPIs as they are the metrics used in day-to-day operations for performance evaluation purposes. By considering the KPIs, the DM verified that the results were satisfying and highlighted the usefulness of this approach in his daily operations.

The rest of the paper is organized as follows. First, some background information of multiobjectove optimization is given in section "Background on multiobjective optimization", while the proposed decision support approach DESMILS to solve a multi-item lot sizing problem is described in section "DESMILS: decision support for a multi-item lot sizing problem". Our case study and the obtained results are described in section "Case study". Finally, conclusions and future research ideas are given in section "Conclusions".

Background on multiobjective optimization

Basic concepts

We consider multiobjective optimization problems formulated as follows:

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

where $k \geq 2$ is the number of objective functions. The objective functions $f_i : S \rightarrow \mathbb{R}$, $i = 1, \ldots, k$, which are at least partly conflicting with each other, are to be optimized simultaneously. The set $S \subseteq \mathbb{R}^n$ is the feasible region formed by constraints. A vector of decision variables $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in S$ is called a feasible solution and the corresponding vector $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$ is called a feasible objective vector, which belongs to the feasible objective region $Z = \boldsymbol{f}(S) \subseteq \mathbb{R}^k$.

In consequence of the conflicting objective functions, multiobjective optimization problems (1) do not typically have any solution where all objective functions can achieve their individual optima. Instead, there are several so-called Pareto optimal solutions that represent trade-offs among the conflicting objective functions. A decision variable vector $\boldsymbol{x}'$ and the corresponding objective vector $\boldsymbol{z}' = f(\boldsymbol{x}')$ are Pareto optimal if there does not exist any $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{x} \in S$ such that $z_i \leq z'_i$ for $i = 1, \ldots, k$ and $z_j < z'_j$ for at least one $j = 1, \ldots, k$. We define an ideal point $\boldsymbol{z}^{\boldsymbol{*}}$ and a nadir point $\boldsymbol{z}^{nad}$ of problem (1) which represent the lower and upper bounds of the ranges of the objective function values among the Pareto optimal solutions, respectively. We also define a vector that is strictly better than the ideal point, which is called a utopian point $\boldsymbol{z}^{\boldsymbol{**}} = (z_1^{**}, \ldots, z_k^{**})^T$ where $z_i^{**} = z_i^* - \epsilon$, $i = 1, \ldots, k$ and ϵ is a relatively small positive scalar.

As the final solution of problem (1), one of the Pareto optimal solutions needs to be selected. The expertise of the DM, who has knowledge about the problem and is responsible for making decisions in the problem domain, is needed in this process. Solving a multiobjective optimization problem means helping the DM in finding his/her most preferred solution. Besides the DM, solving a multiobjective optimization problem involves an analyst. The analyst supports the DM in the mathematical aspects of the problem and is responsible for making preparations of the multiobjective optimization method before the DM is involved.

Many methods have been developed to solve multiobjective optimization problems and they can be classified based on how the DM's preferences are considered in the methods (Miettinen, 1999). No-preference methods do not use any preferences from the DM, a priori methods ask the DM's preferences before running the optimization algorithm, a posteriori methods ask the DM's preferences after having found a representative set of Pareto optimal solutions, and interactive methods ask the DM's preferences iteratively during the decision making process. Among these methods, interactive methods are regarded as promising because they allow the DM to learn during the decision making process and change his/her preferences until he/she finds the best solution for him/her (Miettinen & Mäkelä, 2006; Xin et al., 2018).

Scalarizing functions

Many methods suggested for solving multiobjective optimization problems utilize scalarizing functions (Miettinen, 1999). Via scalarizing functions, the multiple objective functions are transformed into a single objective function and the resulting problem is solved with an appropriate single objective optimization method. Scalarizing functions must be selected carefully, e.g., to guarantee the Pareto optimality of the solution obtained. The scalarizing functions typically include preference information obtained from the DM. There are many ways to ask this information (Miettinen, 1999). One of them is asking for desirable values for each objective function $\tilde{z}_1, \ldots, \tilde{z}_k$. They are called aspiration levels. The vector $\tilde{\boldsymbol{z}}$ consisting of aspiration levels is called a reference point.

Several scalarizing functions have been introduced in the literature (Miettinen & Mäkelä, 2002). One of the widely used scalarizing functions is the achievement scalarizing function (ASF) (Wierzbicki, 1980). An ASF finds the closest Pareto optimal solution to the reference point. This function works well both with feasible and infeasible reference points to find a Pareto optimal solution for the multiobjective optimization problem (1). The ASF which is used in DESMILS can be written as follows:

$$\begin{aligned} \text{minimize} \quad & \max_{i=1,\dots,k} \left\{ \frac{f_i(\boldsymbol{x}) - \tilde{z}_i}{z_i^{nad} - z_i^{**}} \right\} + \rho \sum_{i=1}^{k} \frac{f_i(\boldsymbol{x})}{z_i^{nad} - z_i^{**}} \\ \text{subject to} \quad & \boldsymbol{x} \in S, \end{aligned} \tag{2}$$

where $\rho > 0$ is a relatively small scalar that guarantees the Pareto optimality of the solutions to (1) (Miettinen, 1999).

Synchronous NIMBUS method

The synchronous NIMBUS method (Miettinen & Mäkelä, 2006) is an interactive method that has been used in many applications [see e.g., Saccani et al. (2020), Sindhya et al. (2017) and Ruotsalainen et al. (2010)]. We summarize it here since it will be applied in the case study. In this method, the DM gives her/his preferences with a so-called classification and several scalarizing functions are formulated by using the preference information from the DM to get new Pareto optimal solutions following the preferences.

NIMBUS needs a starting point (a Pareto optimal objective vector), and the DM gives his/her preferences to indicate what kind of changes in the objective function values would lead to a more preferred solution. The starting point can be specified by the DM or it can be a so-called neutral compromise solution which is located, roughly speaking, approximately in the middle of the Pareto optimal set. The neutral compromise solution is calculated by solving the ASF (2) with $\tilde{z}_i = (z_i^{nad} + z_i^{**})/2$ as aspiration levels for $i = 1, \dots, k$. The starting point is presented to the DM in the first iteration, together with the ideal and nadir points. Then, in each iteration, the DM gives his/her preferences by classifying each objective function (with the current value) into up to five classes by indicating whether he/she wants to:

1. improve the current value ($I^{<}$),
2. improve the current value to a certain aspiration level ($I^{\leq}$),
3. keep the current value ($I^{=}$),
4. impair the current value until a certain bound ($I^{\geq}$), or
5. let the current value change freely ($I^{\diamond}$).

When a classification is feasible (i.e., some objective functions are to be improved and some are allowed to get worse), up to four different scalarizing functions are utilized to generate new Pareto optimal solutions reflecting the DM's preferences as well as possible. The DM gives an upper bound for how many solutions he/she wants to see and compare. The new Pareto optimal solutions are then presented to the DM who chooses one solution to continue to the next iteration (use it as the starting point of a new classification) or stop with this solution as the final one, if he/she is satisfied with it. There is also a possibility to generate a desired number of intermediate solutions between any two Pareto optimal solutions. Further details about the synchronous NIMBUS method can be seen in Miettinen and Mäkelä (2006).

DESMILS: decision support for a multi-item lot sizing problem

The idea of DESMILS is to extend a single-item multiobjective lot sizing model to be applied in multi-item lot sizing with a large number of items. This approach can be implemented in any variant of a single-item lot sizing problem, which is intended to be extended to a multi-item problem, if the single-item problem is modeled as a multiobjective optimization problem. As examples, this approach is appropriate for the lot sizing problem under demand uncertainty in Kania et al. (2022), the lot sizing problem with safety stock and safety lead time in Kania et al. (2021), and the lot sizing problem with supplier selection in Ustun and Demirtas (2008). DESMILS enables single-item lot sizing models to be used in case of a large number of items without having to conduct the decision making process separately for every single item.

As said, in multiobjective optimization, the final solution depends on preference information provided by the DM during the decision making process. If the decision making process is considered separately for each item, the DM may provide different preferences in deciding lot sizes for different items. However, repeating the decision making process for each item is laborious in case of a large number of items. To address this concern, we propose a decision support approach that can accommodate item-specific preference information from the DM without a need of repeating the decision making process for each item separately. Here, we refer to item-specific preference information as the preference information that the DM provides for solving a single-item lot sizing problem for a specific item. The proposed approach is called DESMILS as an abbreviation of Decision Support for Multi-Item Lot Sizing Problem.

Considering a large number of items, the DM typically does not have totally different item-specific preference information for all the items. He/she may have similar preferences for some items. He/she usually gives his/her preference information in the lot sizing problem based on some properties, such as price, demand, size, and/or location of the supplier. For example, he/she avoids holding stocks for expensive or large items but carries more stocks (for instance in safety stock) for the items with a high demand. DESMILS divides

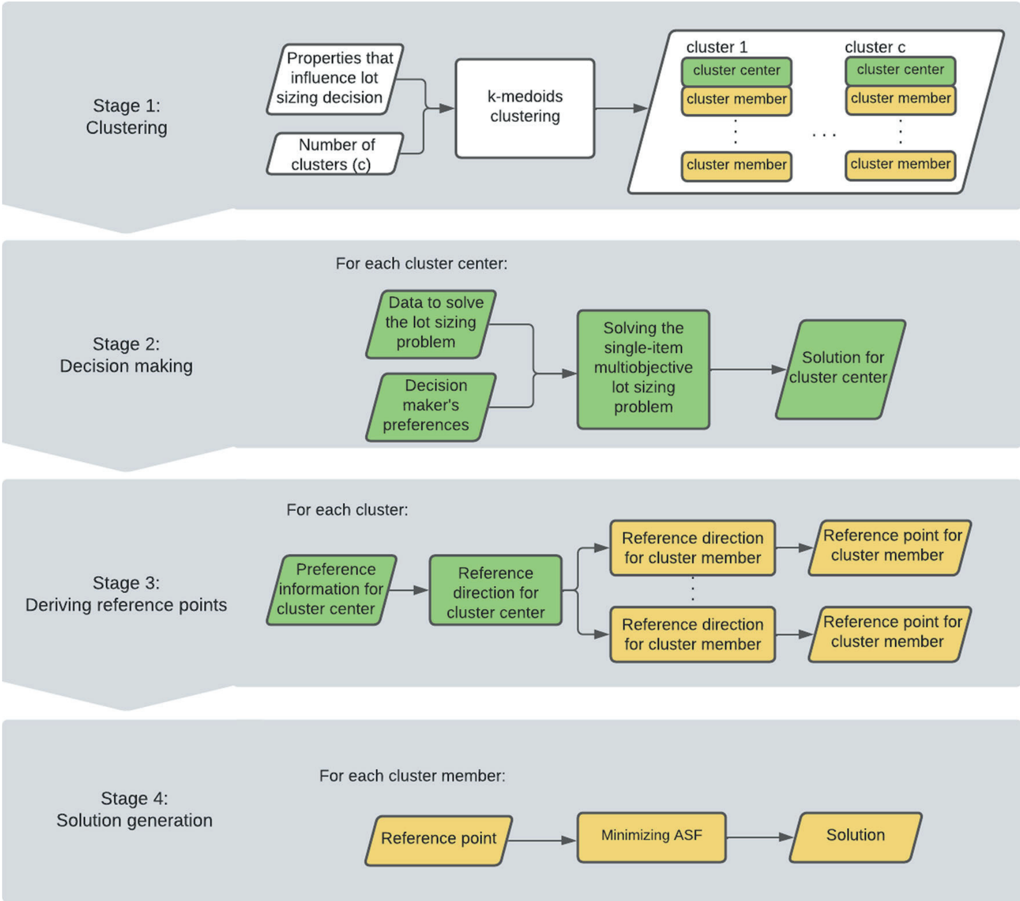


Fig. 1 Flowchart of DESMILS

the items into clusters based on the properties that influence the DM's opinion in making lot sizing decisions. In this way, we assume that the items in the same cluster have similar item-specific preference information, and therefore, the DM only needs to give preference information for one item which is representative of the cluster. Then, this information is extended to other items in the same cluster that are similar enough to the representative one.

DESMILS has four stages, as shown in Fig. 1. We assume that the total number of items is m. In the first stage, these items are divided into c clusters, where c is clearly smaller than m. Each cluster has one or more items with one item regarded as the representative of the cluster. The representative of each cluster is called a *cluster center*. In the second stage, the decision making process is conducted c times with an interactive method, where the DM gives his/her preferences to find preferred lot sizes for each cluster center. The remaining items in the cluster are called *cluster members*. We propose an approach in the third stage to find reference points for these items by using the preference information that the DM provided for the corresponding cluster center and repeat this for each cluster. Finally, we obtain the solutions for the cluster members using these reference points in the last stage.

The involvement of a DM is needed in the clustering stage and the decision making stage. In the clustering stage, the DM is asked to provide the number of decision making processes he/she wants to conduct and check the clustering results. In the decision making stage, the DM provides his/her preferences to solve the single-item lot sizing problem for c cluster centers. The other stages do not involve the DM. There are two kinds of data needed in DESMILS: properties that influence lot sizing decisions, and data needed as input for solving single-item lot sizing problems. For example, in the case study considered in section "Case study", properties that influence lot sizing decisions are SS, SOT, purchasing price, transit time, daily average demand, and physical size of the item. Furthermore, demand data for 24 periods, price, lead time, previous order data, minimum order quantity and rounding value are the input data used to solve the single item lot sizing problems in the case study, where the company needs to solve a multiobjective lot sizing problem described in Appendix A.

In what follows, we give details of each stage.

Clustering stage

As said, the DM's lot sizing decisions are usually influenced by certain properties, and they are used in this stage to divide items into clusters. Therefore, investigating the DM's reasoning in making his/her decision is important in this stage to ensure items with similar item-specific treatment are placed in the same cluster. The analyst can interview the DM to investigate which properties influence his/her lot sizing decisions.

The purpose of the clustering stage is to assign m items into c clusters so that the items in the same cluster can be treated with similar preferences. By using the properties that influence the DM's lot sizing decisions, we divide items into clusters, where each cluster has one representative item as a cluster center and the remaining items as cluster members. Naturally, any appropriate clustering technique, which is usually used in machine learning, can be used in this stage. However, it is important to select a clustering technique that provides one of the items as the center of the cluster and not, for example, some average. Therefore, in this research, we use the k-medoids clustering technique (Kaufman & Rousseeuw, 1990). The idea of taking an item which is nearest to the means of items as the center of the corresponding cluster fits our purpose.

In some clustering methods, including k-medoids, the number of clusters c is required to be specified as input. This enables the DM to decide the number of the decision making processes that he/she prefers to do. The methods that have been developed to determine the optimal number of clusters, such as the elbow method (Thorndike, 1953), which is the oldest and most widely used method in cluster analysis, can also be used to give a suggestion to the DM. However, the number of clusters needs to be confirmed by the DM and the items of each clusters need to be checked by the DM so that items in the same cluster can be treated similarly.

Decision making stage

In the previous stage, c cluster centers were identified to represent all the other items. Therefore, we need to conduct c decision making processes in this stage to solve the single-item lot sizing problem for each cluster center. The data used in this stage depends on the single-item lot sizing problem to be solved.

Any appropriate multiobjective optimization methods can be applied to find the most preferred lot sizes for each cluster center. However, to be able to reflect the preference information from the DM to be used for the next stage, the method used in this stage should have a starting point. In the case study considered in this paper, we used the interactive NIMBUS method as its type of providing preference information was preferred by the DM in question. In NIMBUS, we used a neutral compromise solution (as defined in Sect. 2.3) as a starting point, which helps us to reflect the preference information from the DM to be used for the next stage. The final solutions and the starting points for each cluster center are output of this stage and they are needed in the next stage.

Deriving reference points stage

After obtaining solutions for all cluster centers in the previous stage, we need to determine optimal lot sizes for all cluster members by utilizing the preference information that the DM provided for the corresponding cluster center. In this stage, we derive a reference point for each cluster member and use them to obtain the solution in the next stage. The reference point represents the desired values that the DM wants to achieve for each objective function based on his/her preference information for the cluster center. Since DESMILS repeats the same task for each cluster, in what follows, we describe the solution process for one cluster as an example.

The preference information from the DM is interpreted as the direction from the starting point to the most preferred solution that the DM selected for the cluster center. We call it a reference direction. Figure 2 illustrates the idea how to use this reference direction to get a reference point for one cluster member (the reference points for other cluster members are obtained in the same way). A starting point for the cluster member is needed and it can be calculated in the same way

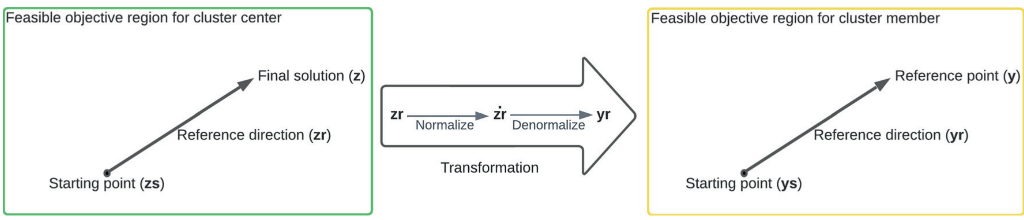


Fig. 2 The idea of finding a reference point to obtain the solution for the cluster member

as in the interactive method that was used in the previous stage. By moving from the starting point in the direction of the reference vector, a reference point for the cluster member is obtained.

We need to emphasize that each item has its own set of Pareto optimal solutions, which means that the cluster center and cluster member have different feasible objective regions. Therefore, transformation is needed to make the reference direction of the cluster center appropriate for the cluster member. For this purpose, we first normalize the reference direction of the cluster center to a proportional position, and then denormalize the proportional position of the reference direction to the region of the cluster member. After the normalization and denormalization processes, the reference direction can be used to find a reference point for the cluster member. Algorithm 1 outlines the general idea of this stage and the details of the algorithm are given afterwards.

Algorithm 1: Algorithm to derive reference points for each cluster member

Input: The starting point of the cluster center $\boldsymbol{zs}$ and final solution of the cluster center $\boldsymbol{z}$
Output: The reference point for each cluster member
Calculate the reference direction for the cluster center $\boldsymbol{zr}$
Normalize $\boldsymbol{zr}$ to a proportional position $\dot{\boldsymbol{zr}}$
foreach *cluster member* **do**
    Calculate the starting point of the cluster member $\boldsymbol{ys}$
    Denormalize $\dot{\boldsymbol{zr}}$ into the feasible objective region of cluster member, denoted by $\boldsymbol{yr}$
    Calculate the reference point $\boldsymbol{y}$
end

From the previous stage, for the cluster center, the starting point $\boldsymbol{zs}$ and the final solution $\boldsymbol{z}$ have been obtained. They are used to calculate the reference direction for the cluster center $\boldsymbol{zr} = (zr_1, \ldots, zr_k)^T$, where $zr_i = z_i - zs_i$, $i = 1, \ldots, k$. This reference direction is then normalized to a proportional position $\dot{\boldsymbol{zr}} = (\dot{zr}_1, \ldots, \dot{zr}_k)^T$ using the following formula:

$$\dot{zr}_i = \frac{zr_i}{zs_i}, \quad i = 1, \ldots, k.$$

To avoid the division by zero, when $zs_i = 0$ for at least one i, the feasible objective region can be shifted, for example, by one unit. This means that one unit is added to all values of the reference direction and the starting point ($zr_i = zr_i + 1$ and $zs_i = zs_i + 1$ for $i = 1, \ldots, k$).

The normalized reference direction $\dot{\boldsymbol{zr}}$ is utilized for all cluster members in this cluster to find a reference point for each cluster member. In what follows, we describe the process to find the reference point for one member, as an example.

The starting point for the cluster member, denoted by $\boldsymbol{ys} = (ys_1, \ldots, ys_k)^T$, is calculated in the same way as in the second stage for the cluster center. The reference direction for the cluster member $\boldsymbol{yr} = (yr_1, \ldots, yr_k)^T$ is then calculated by denormalizing $\dot{\boldsymbol{zr}}$ into the feasible objective region of the cluster member using the following formula:

$$yr_i = \dot{zr}_i \; ys_i, \quad i = 1, \ldots, k.$$

To find the reference point for the cluster member, the starting point $\boldsymbol{ys}$ is directed to follow the preference information from the DM which is represented in the reference direction $\boldsymbol{yr}$. The reference point $\boldsymbol{y} = (y_1, \ldots, y_k)^T$ is then obtained with the following formula:

$$y_i = yr_i + ys_i, \quad i = 1, \ldots, k.$$

Solution generation stage

Reference points found in the previous stage represent the preferred solutions that the DM wants to achieve for each cluster member. However, $\boldsymbol{y}$ may not be a Pareto optimal solution of the lot sizing problem of the cluster member. Therefore, we find the closed Pareto optimal solution by minimizing the ASF(2) with $\boldsymbol{y}$ as the reference point. In this way, a Pareto optimal solution which represents the DM's preference is found for each item.

Case study

In this section, we demonstrate how the proposed approach DESMILS can provide decision support in solving a real lot sizing problem in a manufacturing company. To be more specific, the company is a semi-heavy vehicles company. The company considered needed to determine the optimal lot sizes for 94 items. From the ERP system of the company, we received two kinds of data needed in DESMILS: properties that influence lot sizing decisions, and data needed as input for solving single-item lot sizing problems.

The company deals with a multi-item lot sizing problem within periodic review policy under stochastic environment on demand. To handle demand uncertainty, they hold extra stock with the combination of safety stock (SS) and safety order time (SOT). For performance measurement, the company uses KPIs. Among these KPIs, they selected purchasing and ordering costs (POC), holding cost (HC), cycle service level (CSL) and inventory turnover (ITO) as the most important KPIs for lot sizing decisions. They found the multiobjective lot sizing model described in Kania et al. (2022) to best match their needs, where their KPIs are objective functions to be optimized. Thus, the model has four objective functions: minimizing POC, minimizing HC, maximizing CSL and maximizing ITO. Details of the multiobjective optimization problem, which is solved in this section, are given in Appendix A.

In this case, the time period for inventory planning was one week, and the company wanted to determine the optimal order quantity for 24 weeks and simultaneously decide the optimal values of SS and SOT. In the beginning of each period, the company needs to place an order for each item, and the order arrives after a constant lead time. The company has agreements with suppliers limiting the orders: they are only able to order at least a certain minimum order quantity and multiples of a rounding value. The minimum order quantities, rounding values, and lead times vary for different items and these are specified as input of the optimization problem. Besides that, the predicted demand data for the following 24 weeks, the previous orders that are supposed to arrive during the lead time period, the price to purchase one unit of item, and the cost to place an order were also needed as input of the optimization problem (see Appendix A).

The supply chain manager of the company is responsible for making lot sizing decisions and he was the DM in this study. He agreed with the model described in Appendix A, but wanted to add bounds for CSL and ITO as additional constraints. The minimum value of CSL which was acceptable for him was 0.9. For ITO, the DM appreciated high value but values higher than 80 were not reasonable for him.

Clustering stage

First, we interviewed the DM to understand which properties influence his decisions in lot sizing. The DM said that there are six relevant properties: SS, SOT, purchasing price, transit time, daily average demand, and physical size of the item. SS and SOT are the results of optimization, but the company predicts them for production planning purposes and they are used by the DM to set desired values for CSL. The purchasing price is important in deciding POC and HC, transit time influences his desires in CSL and ITO, while daily average demand is necessary for all objective functions. To consider the physical size of an item, the DM has access to data on the 'number of units in one handling unit'. It shows the number of units of an item that can be packed in one handling unit, for example, a pallet. One handling unit can store many units of an item if it is a small item, otherwise, it is only able to store few units of a big item. This data affects his decisions in deciding HC and CSL.

As said, we received data from the ERP system of the company containing information about the six properties that influence the DM's lot sizing decisions. The data was used to cluster the 94 items with the k-medoids clustering technique. To help in determining the number of clusters, an elbow graph was presented to the DM showing the distortion of the sum of square error values of the distances between cluster centers and cluster members. The best number of clusters is usually found if there is an 'elbow' in the curve, that is, where the distortion of the following cluster does not decrease much. However, in this case, the distortion basically decreased when the number of cluster increased, but there was no elbow visible. Therefore, the decision of the number of clusters relied on the DM.

According to the DM, an acceptable number of clusters for 94 items was between 7 and 12 clusters. Therefore, he wanted to see the clustering results in this range (i.e, cluster centers and cluster members for different numbers of clusters). After comparing the clustering results of 7–12 clusters, the DM decided that the appropriate number of clusters was 10. The reason was that with 10 clusters, the items in the same cluster could be best treated with similar preferences. The result of the clustering with 10 clusters is presented in Fig. 3, where different colours represents different clusters. Therefore, the DM needed to complete a total of 10 decision

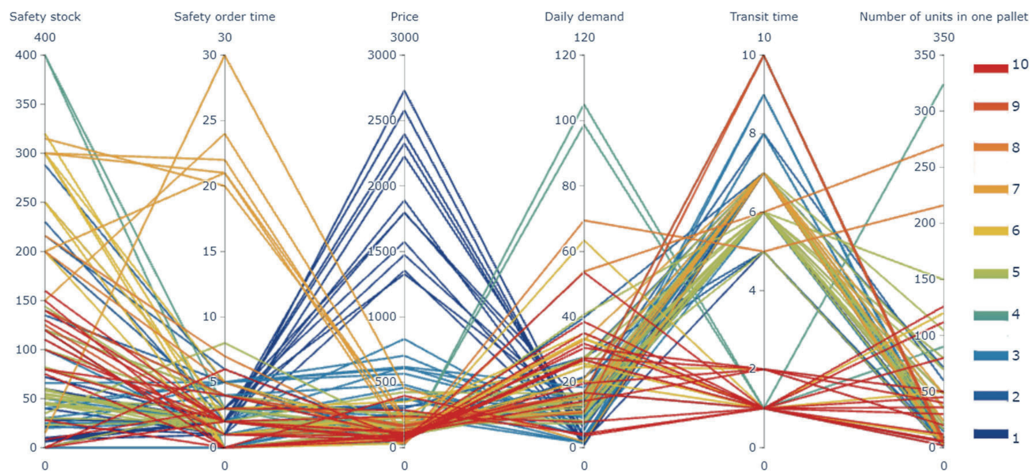


Fig. 3 Result of clustering (ten clusters indicates by different colours)

making processes, and this number was acceptable for him. (This number is clearly lower than repeating the process for each of the 94 items.)

Decision making stage

For compactness, we here describe the solution process for one cluster only (the other clusters were treated in the same way). The considered cluster is shown in red colour in Fig. 3 (cluster 10). Items in this cluster have low purchasing prices, low transit times, and quite low values for the other elements. Based on the data from the company, the cluster center of this cluster has the price of 57.09 and the lead time of four weeks. The minimum order quantity and the rounding value of this item are both 45 units, while the demand data and the previous orders that are supposed to arrive during the four week lead time period, can be seen in Fig. 4.

The DM wanted to use the interactive NIMBUS method to find the best lot sizes for the cluster centers since he preferred to give his preferences in the form of a classification and he loved the way NIMBUS handles classification. However, the lot sizing problem to be solved is computationally expensive (Kania et al., 2022), and therefore solving one scalarizing function spends several minutes and NIMBUS needs to solve up to four scalarizing functions on each iteration. To reduce the waiting time of the DM, we generated a representative set to approximate Pareto optimal solutions in advance, and used NIMBUS to help the DM select one of them.

Because of the complexity of lot sizing problems, evolutionary algorithms, have become popular and efficient tools to approximate the set of Pareto optimal solutions in these problems (Goren et al., 2010). In this case, we applied an evolutionary method called NSGA-III (Deb & Jain, 2014), which has been developed for multiobjective optimization problems with more than three objective functions. We applied the implementation of NSGA-III in a framework called pymoo (Blank & Deb, 2020), because it can handle integer variables and many constraints. Details of generating the representative set for the cluster center are presented in Appendix B.

A graphical user interface is important in decision making processes with interactive methods to facilitate interaction between the DM and the method. We used DESDEO (Misitano et al., 2021), an open source Python framework, which provides implementations and graphical user interfaces for various interactive multiobjective optimization methods, including NIMBUS. The feature of having a pre-generated set of solutions is also provided in this framework.

As mentioned in Sect. 2.3, in the first iteration of NIMBUS, the starting point together with the ideal and nadir points are presented to the DM to support providing the first classification. Figure 5 shows the corresponding screenshot of NIMBUS in DESDEO. In this case, the starting point (objective vector) for the cluster center was (146 066.8, 525.1, 0.98, 44.65), while the ideal and nadir points were (144 466.8, 332.42, 1, 79.42) and (152 604.9, 2 989.05, 0.906, 9.89), respectively. The objective function values in the starting point are indicated by pink bars in Fig. 5. The graphical user interface supports the DM in remembering the direction of improvement. The first and the second objective functions are to be minimized (pink bar starts from the left) and the others are to be maximized (pink bars start from the right); and the shorter the pink bar, the closer the current value is to the ideal value.

In the first iteration, the DM wanted to improve ITO until 60, and allowed CSL to decrease until 0.91, while the other objective functions were allowed to change freely. He wanted to compare up to four solutions, but he only got two different solutions because of the same results in optimizing some of the scalarizing functions. The solutions were (147 266.8, 332.42, 0.9258, 79.42) and (147 266.8, 361.61, 0.9747, 72.37). The solutions were visualized for the DM in DESDEO to help comparisons. The DM chose the second solution since it had a better CSL value. The ITO value of this solution was worse than in the first one, but it was acceptable for the DM. The DM continued to the next iteration with the selected solution.

The DM was already rather satisfied with the current solution, but he wanted to explore whether he could get a better solution. (He appreciated the feature of NIMBUS that allowed him to go back to the previous solution if the solutions of the next iterations are not getting better. Thus, there was no risk of losing the previous solution by trying new preferences.) For the second iteration, he allowed to impair ITO until 25, but he wanted to improve CSL until 0.99 and let the other objective functions change freely. The solutions obtained in this iteration were (151 004.9, 729.45, 0.999999995, 26.92), (146 866.8, 455.03, 0.996, 52.55), (149 035.85, 615.6, 0.999997, 32.7) and (152 604.9, 565.97, 0.999999994, 34.67). The DM selected the second solution, where he got the best values for POC, HC and ITO, and the CSL value was acceptable. When compared to the solution of the first iteration, the current solution had a better CSL value and an acceptable value for ITO, hence the DM decided to continue with the current solution for the next iteration.

The DM was satisfied with the CSL and ITO values and wanted to improve HC as much as possible in this iteration. Because of trade-offs, he had to allow impairment in at least one other function, and he preferred to sacrifice ITO a bit until 50. He allowed POC to change freely and kept CSL in the current value. He wanted to see up to four solutions but he only got three different ones. The solutions were (150 035.85, 405.4, 0.997, 55.91), (147 266.8, 332.42, 0.926, 79.42) and (147 466.8, 390.81, 0.995, 58.51). He selected the first one with the best CSL value and an acceptable ITO value. He was planning to stop with this solution. However, when he saw the corresponding decision variable values, he found SS and

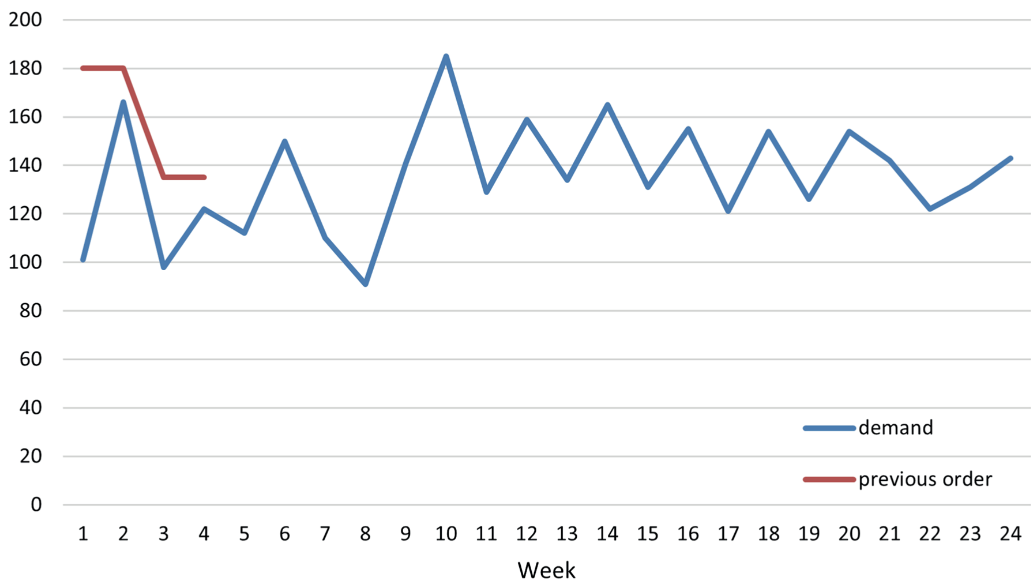


Fig. 4 Demand and previous order data for the cluster center

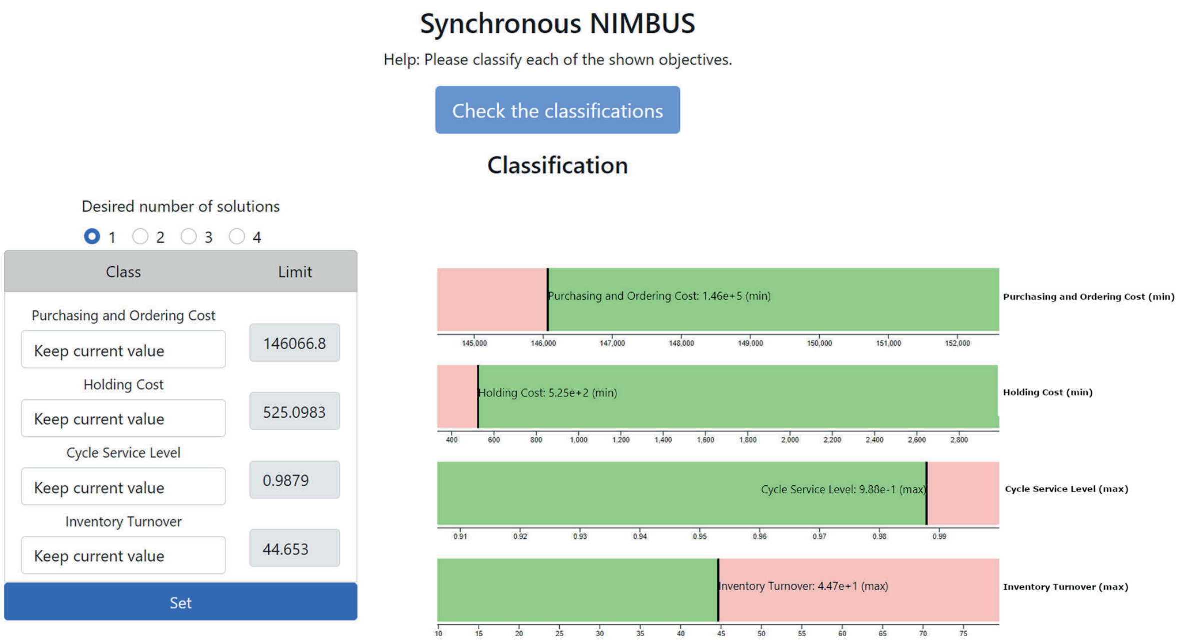


Fig. 5 Graphical user interface of NIMBUS in DESDEO

SOT values unacceptable, and wanted to start the decision making process again from the beginning to get a better CSL value.

The DM was again shown the information in Fig. 5. He wanted to improve CSL until 0.9999, sacrifice ITO to 40 and let POC and HC change freely. He wanted to see up to four solutions but got these two solutions: (150 035.85, 498.82, 0.999993, 40.89). Based on the previous experiences and learning there, he wanted to play safe with CSL and chose the first one with a better CSL value.

In the second iteration, he preferred to improve HC until 400 and sacrifice on ITO to 35. He let POC change freely and kept CSL in the current value. He was then presented with these four solutions: (150 035.85, 498.82, 0.999993,

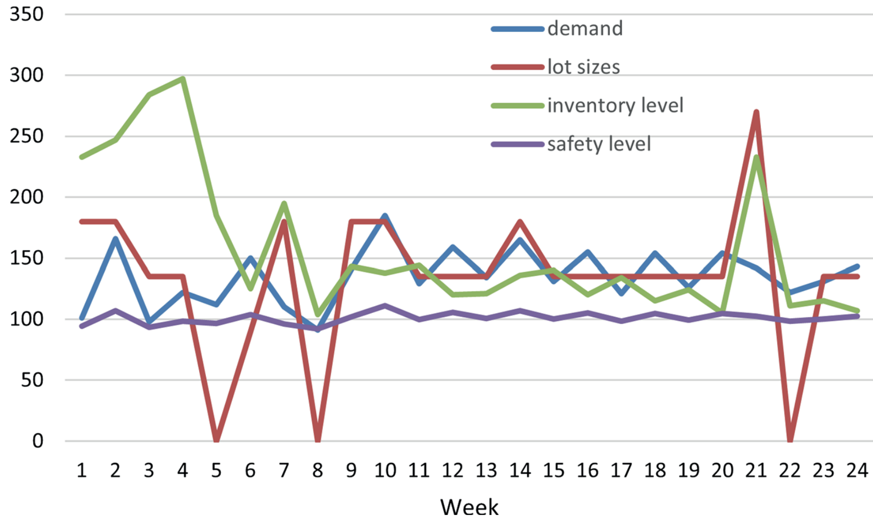


Fig. 6 Result for cluster center

40.89), (149 835.85, 492.99, 0.999991, 41.13), (149 835.85, 411.24, 0.997, 55.02) and (150 035.85, 405.4, 0.997, 55.91). The second solution was the best for the DM and he decided to stop with it as the final one. The DM was very pleased with the final solution as well as the corresponding decision variable values.

The lot sizes that arrive for each planning period can be seen in Fig. 6 in red. The previously set order data for the first four weeks are followed by the optimized lot sizes after week 4 (the lead time was 4 weeks in this cluster). The figure shows that no order is needed for weeks 5, 8, and 22. Following the DM's preferences in the decision making process, we do not need to order in every single period to have a balance between POC and HC. In this case, orders for weeks 5, 8, and 22 are unnecessary to save on ordering costs. The final SS and SOT values were 74 units and one day, respectively. The inventory level indicated by the green line shows that the company had excess inventory during the lead time period, and it then decreased and followed the demand quantity with the optimized lot sizes. Thus, the company saved money invested in the inventory. The DM was pleased with the improvement in the inventory level but keep the safety level high, following his preferences.

Deriving reference points stage

From the previous stage, we got the final, optimized solution for the cluster center $z = (149\,835.85, 492.99, 0.999991, 41.13)$ while the starting point of the interactive solution process was $zs = (146\,066.8, 525.1, 0.98, 44.65)$. With these points, we calculated the reference direction of the cluster center as $zr = (3\,769.05, -32.11, 0.012132, -3.53)$ and the normalization of zr was $\dot{zr} = (0.0258, -0.0612, 0.01228, -0.07896)$.

This cluster had 14 cluster members (besides the cluster center). As described in section "Deriving reference points stage", we calculated starting points for each cluster member. Because we used NIMBUS and the neutral compromise solution as the starting point for the cluster center, we calculated neutral compromise solutions as starting point for cluster members. We then followed Algorithm 1 to calculate the reference point for each cluster member. The starting points and the reference points for the cluster members in this cluster can be seen in Table 1.

Solution generation stage

For each cluster member, we considered the corresponding reference point, minimized the ASF (2) and derived a solution. These solutions are presented in Table 2. The DM accepted them and appreciated that each item had its solutions following his preferences. He was able to find solutions for the cluster with 15 items with only one decision making process, thanks to DESMILS.

The steps from the decision making stage until the solution generation stage were repeated for other clusters. The DM provided different preferences in the decision making process for the different cluster centers and he was pleased with the results of both cluster centers and cluster members, which followed his preferences.

Compared with the traditional method used in the company (without any decision support tool), the DM emphasized the following benefits in using DESMILS.

1. The DM can consider different KPIs simultaneously and understands the trade-offs among them, when he is able to compare different solutions and change his preferences during the decision making process. Thus, he can train his

Table 1 Starting points and reference points for cluster members

Item	Starting points				Reference points			
	POC	HC	CSL	ITO	POC	HC	CSL	ITO
1	202 451.84	1 778.92	0.912	21.68	209 396.1	554.38	0.924	54.79
2	110 688.96	1 232.14	0.965	13.99	114 485.68	383.98	0.978	35.38
3	248 950	2 875.84	0.926	13.34	257 489.19	896.22	0.938	33.73
4	142 384.4	1 570.19	0.999	12.96	147 268.3	489.33	1.014	32.75
5	216 751.6	2 724.29	0.922	12.83	224 186.36	848.99	0.935	32.44
6	256 988.32	2 483.68	0.977	19.08	265 803.23	774.01	0.991	48.24
7	178 282	1 229.85	0.939	21.63	184 397.22	383.27	0.952	54.68
8	158 921	2 510.89	0.906	10.61	164 372.12	782.49	0.919	26.82
9	139 796	2 495.98	0.923	11.65	144 591.12	777.84	0.935	29.45
10	296 462.4	4 069.06	0.956	13.45	306 631.3	1268.08	0.969	34.01
11	166 792.35	1 597.38	0.914	17.46	172 513.46	497.81	0.927	44.14
12	172 367.85	1 873.49	0.902	16.24	178 280.21	583.85	0.915	41.05
13	214 529.6	2 672.44	0.965	14.83	221 888.14	832.84	0.978	37.49
14	176 194.6	1 896.18	0.933	15.46	182 238.22	590.92	0.946	39.09

Table 2 Solutions for cluster members

Item	POC	HC	CSL	ITO
1	205 331.2	330.48	0.939	77.92
2	112 688.96	458.07	0.973	38.13
3	251 550	690.96	0.942	78.21
4	144 784.4	652.87	0.999	32.79
5	221 748.84	591.29	0.947	66.14
6	265 234.88	571.19	0.990	62.16
7	180 882	524.75	0.951	55.34
8	161 321	870.53	0.950	50.54
9	142 589.12	341.04	0.945	60
10	298 862.4	1 071.85	0.975	64.02
11	169 749.9	482.91	0.953	79.99
12	175 525.4	397.89	0.945	79.99
13	220 466	515.15	0.991	64.8
14	179 509.68	539.97	0.956	79.96

team members and other stakeholders of the company on this aspect of lot sizing for better results.

2. The optimal lot sizes provided by DESMILS improve inventory planning and control in his company. The inventory value, which is a core KPI for the top management, was reduced for all items in this case study.
3. Saving time is a significant issue in daily operations. Compared with the previous way, where it is mostly done item by item, DESMILS save a significant amount of time and effort. DESMILS also allows the DM to decide the number of clusters, and therefore, he can control the effort needed to solve his multi-item lot sizing problems.
4. DESMILS also reduces the risk of human error. When processes are not controlled only by traditional methods, the risk of unintentional forgetting is reduced. It in turn supports production needs when the right amount of material is available at the right time.

As said, the company already had KPIs in use, and the suitable ones were selected as the objective functions. In this way, the results of the optimization were used as a source of information to KPIs, for example, for reporting purposes to senior management. Based on the KPI information in the objection functions, he confirmed that the results are acceptable and reflect his preferences well. This allows him to focus on nurturing and developing company's buyer–supplier relationships and developing lot sizing processes there.

Being a good buyer with convincing and predictable lot sizing planning is a good method to successful buyer–supplier relationship when creating competitive advantage. Naturally, our approach does not only focus on the development of the activities of the company in question. Production companies in general could improve their inventory management with our approach.

Conclusions

In this paper, we have introduced DESMILS, a decision support approach to solve multi-item lot sizing problems. Our motivation is to enable any single-item lot sizing model, which is formulated as a multiobjective optimization problem, to be applied in multi-item problems with a large number of items. Our approach applies an interactive multiobjective optimization method to solve a single-item lot sizing problem for few selected items. It then accommodates preferences obtained from the DM so that the DM does not need to repeat the decision making process for each item separately. The preferences are used to derive optimal lot sizes for the other items.

The idea of DESMILS is to divide items into clusters using properties that influence the DM's lot sizing decisions, with the reasoning that items in the same cluster can be treated with similar preferences in the lot sizing decision. Therefore, we only need to conduct the decision making process, where the DM provides his/her preferences, for one representative item for each cluster. We then translate the preference information to derive Pareto optimal lot sizes for the remaining items in the same cluster. In this way, optimal lot sizes that represent the DM's preferences are obtained for all items.

As a proof of concept, a real lot sizing problem from a manufacturing company was solved to demonstrate the applicability of the proposed approach. Lot sizes were to be determined for 94 items and with DESMILS, Pareto optimal solutions reflecting the DM's preferences were found for all items. However, the DM had to solve only a limited number of lot sizing problems. The DM was satisfied with all of the solutions and the corresponding decision variables. He appreciated that he could find lot sizes for each item reflecting his preferences with a limited amount of effort from his side.

Solving multi-item lot sizing problems incorporating a DM's preferences in deciding lot sizes for different items was proposed for the first time in this research. Hence, testing this approach with different types and characteristics of the problems and with different numbers of items are topics of future research extending this work. In our case study, the elbow method failed to help the DM in setting the number of clusters. Therefore, our future work includes finding better support the DM in this. Furthermore, in the case considered, there is no information about connections and dependencies between items, but it can be a possible future research direction.

Acknowledgements This research was partly funded by LPDP, Indonesian Endowment Fund for Education (grant number S-5302/LPDP.4/2020), and the Academy of Finland (grant numbers 322221 and 311877). 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 Jussi Lehtimäki for being the decision maker and providing data and valuable insight in practical point of view. We also thank Giovanni Misitano for his help related to the user interface.

Funding Open Access funding provided by University of Jyväskylä (JYU).

Declarations

Competing interests 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.

Appendix A: Multiobjective optimization model

Based on the needs of the real case study, we used the lot sizing model proposed in Kania et al. (2022) to consider single-item lot sizing. This model follows a periodic review policy, where orders are reviewed over discrete time periods $t = 1, \ldots, T$. This is a single-item lot sizing model to determine the optimal order quantity ($Q(t)$) for each period considered and simultaneously decide the optimal values of SS and SOT. There are four objective functions and four constraints in the model. The two objective functions related to costs (i.e., POC and HC) are considered as different objective functions here, because there is trade-off between them and the DM wants to study the trade-off.

$$
\begin{aligned}
\min \quad & POC = \sum_t Q(t)\, p + \sum_t Y(t)\, c, \\
& HC = \sum_t \frac{I(t-1) + I(t)}{2}\, h, \\
\max \quad & CSL = F\left(\frac{SS + \mu\, SOT}{\sigma}\right), \\
& ITO = \sum_t \frac{D(t) + \sigma}{(I(t-1) + I(t))/2}, \\
\text{s.t.} \quad & \frac{I(t-1) + \sum_{i=t-\lfloor L \rfloor}^{t} Q(i) - SS}{\sum_{j=t}^{t+\lfloor P \rfloor} D(j) + (P - \lfloor P \rfloor) D(\lceil P \rceil)} \geq 1, \\
& \text{for } t = 1, \ldots, T, \\
& Q(t) = Y(t)\,(moq + a\, r) \text{ , for any integer } a \geq 0 \\
& \text{ and } t = 1, \ldots, T, \\
& I(t) \geq SS + SOT\, D(t) \text{ , for } t = 1, \ldots, T, \\
& SS \geq 0 \text{ and } SOT \geq 0,
\end{aligned}
$$

where

p price to purchase one unit of the item
c cost to place one order
h cost to hold one unit for one period
L lead time
$D(t)$ predicted demand during period t

σ standard deviation of demand $D(t)$
μ average demand $D(t)$
moq minimum order quantity (for lot size)
r rounding value (for lot size)
$Y(t)$ order indicator ($Y(t) = 1$ if $Q(t) > 0$, otherwise $Y(t) = 0$)
$I(t)$ inventory position at the end of period t ($I(t) = I(t-1) + Q(t - \lfloor L \rfloor) - D(t)$)
P the consideration period for one order ($P = L + SOT$).

Appendix B: Details of generating solutions for the decision making stage

As said, the lot sizing problem to be solved in the case study is a computationally expensive problem. Therefore, generating many solutions to approximate Pareto optimal solutions is a challenge. The minimum order quantity and rounding value as well as constraints limit the range of feasible solutions. Here, we applied NSGA-III by using the pymoo framework. We combined solutions obtained with different initial populations and various parameters of evolutionary operators that were available in the framework, to get more different solutions (Deb & Jain, 2014).

We applied the structured approach described in Das and Dennis (1998) with the number of partitions from 1 until 20 to generate initial populations. We also combined different types of crossover operators for integer variables, i.e., simulated binary crossover, exponential crossover, uniform crossover, half uniform crossover, and four point crossover. We used crossover probability of 0.9 for all of them, except exponential crossover where we used probability of 0.95. For mutation, we used polynomial mutation for integer variables with mutation probability 0.9. The parameters were selected after several experiments and we found that these parameters were good enough for our case. For other parameters, we used the default values in pymoo (Blank & Deb, 2020). In this way, we obtained 568 solutions for the cluster center in Sect. 4.2 in almost 24 h. However, this process was done without the involvement of the DM, and there was no computational overhead involved in the interactive process.

References

Absi, N., Detienne, B., & Dauzére-Pérès, S. (2013). Heuristics for the multi-item capacitated lot-sizing problem with lost sales. *Computers & Operations Research, 40*(1), 264–272.

Aissaoui, N., Haouari, M., & Hassini, E. (2007). Supplier selection and order lot sizing modeling: A review. *Computers & Operations Research, 34*(12), 3516–3540.

Akyuz, G. A., & Erkan, T. E. (2010). Supply chain performance measurement: A literature review. *International Journal of Production Research, 48*(17), 5137–5155.

Ammar, H. B., Ayadi, O., & Masmoudi, F. (2020). An effective multi-objective particle swarm optimization for the multi-item capacitated lot-sizing problem with set-up times and backlogging. *Engineering Optimization, 52*(7), 1198–1224.

Andriolo, A., Battini, D., Grubbström, R. W., 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.C.N. (2010) Multi-objective optimization for supply chain management: A literature review and new development. In: *Proceedings of the 8th international conference on supply chain management and information systems*, (pp. 425–432). IEEE, Hong Kong

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

Brahimi, N., Absi, N., Dauzère-Pérès, S., & Nordli, A. (2017). Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research, 263*(3), 838–863.

Copil, K., Wörbelauer, M., Meyr, H., & Tempelmeier, H. (2017). Simultaneous lotsizing and scheduling problems: A classification and review of models. *OR Spectrum, 39*, 1–64.

Cunha, J. O., Kramer, H. H., & Melo, R. A. (2019). Effective matheuristics for the multi-item capacitated lot-sizing problem with remanufacturing. *Computers & Operations Research, 104*, 149–158.

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.

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

Goren, H. G., Tunali, S., & Jans, R. (2010). A review of applications of genetic algorithms in lot sizing. *Journal of Intelligent Manufacturing, 21*, 575–590.

Harris, F. W. (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*. https://doi.org/10.1504/IJLSM.2021.10042370

Kania, A., Sipilä, J., Afsar, B., & Miettinen, K. (2021). Interactive multiobjective optimization in lot sizing with safety stock and safety lead time. In M. Mes, E. Lalla-Ruiz, & S. Voß (Eds.), *Computational Logistics, 12th International Conference, ICCL 2021, Proceedings* (Vol. 13004, pp. 208–221). Switzerland: Springer.

Kania, A., Sipilä, J., Misitano, G., Miettinen, K., & Lehtimäki, J. (2022). Integration of lot sizing and safety strategy placement using interactive multiobjective optimization. *Computers & Industrial Engineering, 173*, 108731.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Wiley.

Li, Y., Tao, Y., & Wang, F. (2012). An effective approach to multi-item capacitated dynamic lot-sizing problems. *International Journal of Production Research, 50*(19), 5348–5362.

Mehdizadeh, E., Hajipour, V., & Mohammadizadeh, M. R. (2016). A bi-objective multi-item capacitated lot-sizing model: Two Pareto-based meta-heuristic algorithms. *International Journal of Management Science and Engineering Management, 11*(4), 279–293.

Melega, G. M., de Araujo, S. A., & Jans, R. (2018). Classification and literature review of integrated lot-sizing and cutting stock problems. *European Journal of Operational Research, 271*(1), 1–19.

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

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* (2nd ed., pp. 931–980). Springer.

Miettinen, K., & Mäkelä, M. M. (2002). On scalarizing functions in multiobjective optimization. *OR Spectrum, 24*, 193–213.

Miettinen, K., & Mäkelä, M. M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research, 170*(3), 909–922.

Misitano, G., Saini, B. S., Afsar, B., Shavazipour, B., & Miettinen, K. (2021). DESDEO: The modular and open source framework for interactive multiobjective optimization. *IEEE Access, 9*, 148277–148295.

Naeem, M. A., Dias, D. J., Tibrewal, R., Chang, P. C., & Tiwari, M. K. (2013). Production planning optimization for manufacturing and remanufacturing system in stochastic environment. *Journal of Intelligent Manufacturing, 24*, 717–728.

Ruotsalainen, A., Miettinen, K., & Palmgren, J.-E. (2010). Interactive multiobjective optimization for 3D HDR brachytherapy applying IND-NIMBUS. In D. Jones, M. Tamiz, & J. Ries (Eds.), *New developments in multiple objective and goal programming* (pp. 117–131). Springer.

Saccani, G., Hakanen, J., Sindhya, K., Ojalehto, V., Hartikainen, M., Antonelli, M., & Miettinen, K. (2020). Potential of interactive multiobjective optimization in supporting the design of a groundwater biodenitrification process. *Journal of Environmental Management, 254*, 109770.

Sindhya, K., Manninen, A., Miettinen, K., & Pippuri, J. (2017). Design of a permanent magnet synchronous generator using interactive multiobjective optimization. *IEEE Transactions on Industrial Electronics, 64*(12), 9776–9783.

Tanskanen, K., & Aminoff, A. (2015). Buyer and supplier attractiveness in a strategic relationship: A dyadic multiple-case study. *Industrial Marketing Management, 50*, 128–141.

Thorndike, R. L. (1953). Who belongs in the family? *Psychometrika, 18*, 267–276.

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

Wierzbicki, A. P. (1980). The use of reference objectives in multiobjective optimization. In G. Fandel & T. Gal (Eds.), *Multiple criteria decision making theory and application* (pp. 468–486). Springer.

Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., & Liu, B. (2018). Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access, 6*, 41256–41279.

Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science, 2*, 165–193.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645–678.