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

Published online: 10 April 2023

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 $\boldsymbol{z^*}$ 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 $\boldsymbol{z^{**}} = (z_1^{**}, \ldots, z_k^{**})^T$ 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:

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

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 $i = 1, \ldots, k$. 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

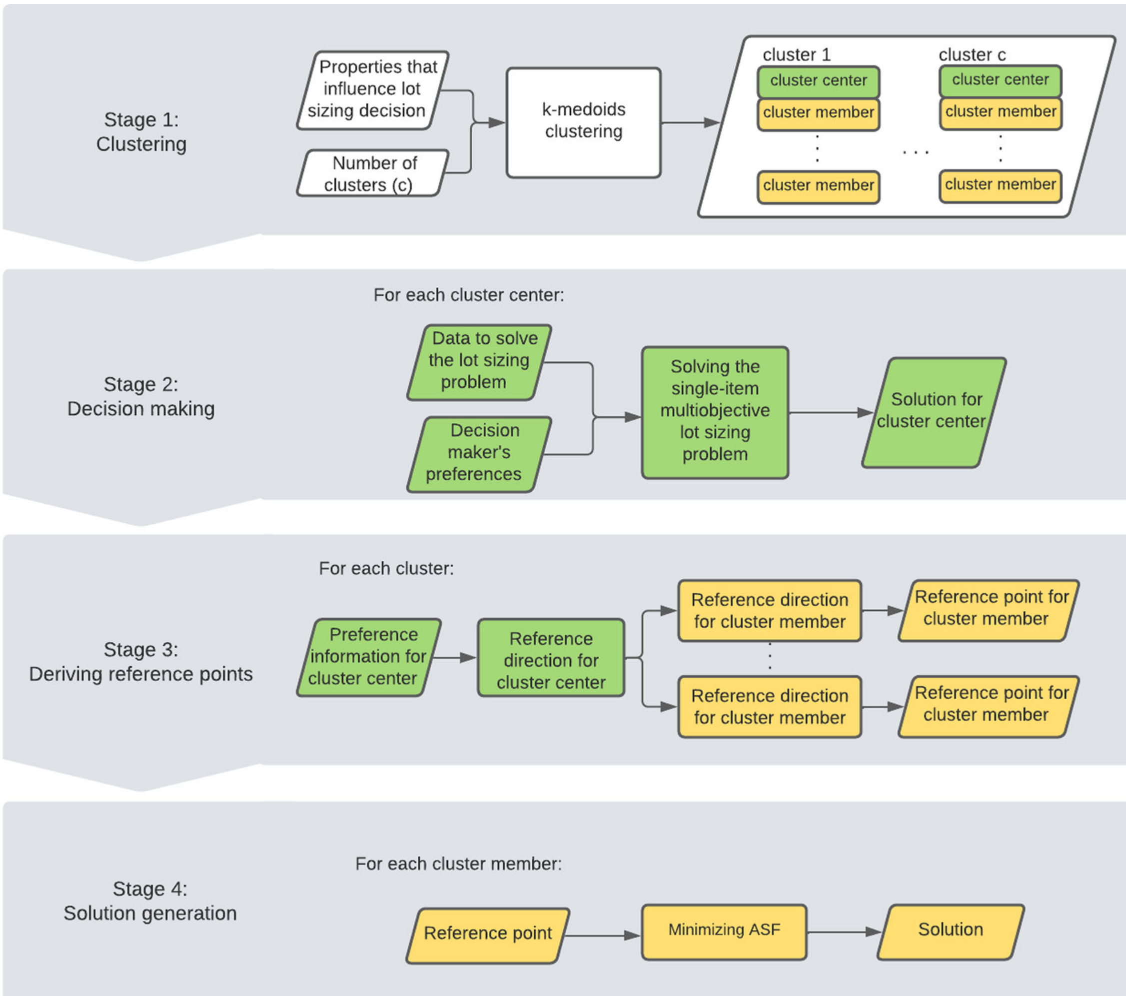

Fig. 1 Flowchart of DESMILS

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

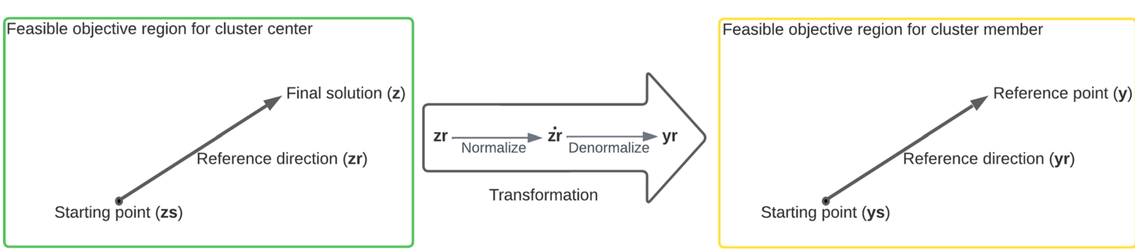

Fig. 2 The idea of finding a reference point to obtain the solution for the cluster member

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 zs and final
       solution of the cluster center z
Output: The reference point for each cluster member
Calculate the reference direction for the cluster center zr
Normalize zr to a proportional position żr
foreach cluster member do
    Calculate the starting point of the cluster member ys
    Denormalize żr into the feasible objective region of cluster
    member, denoted by yr
    Calculate the reference point 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

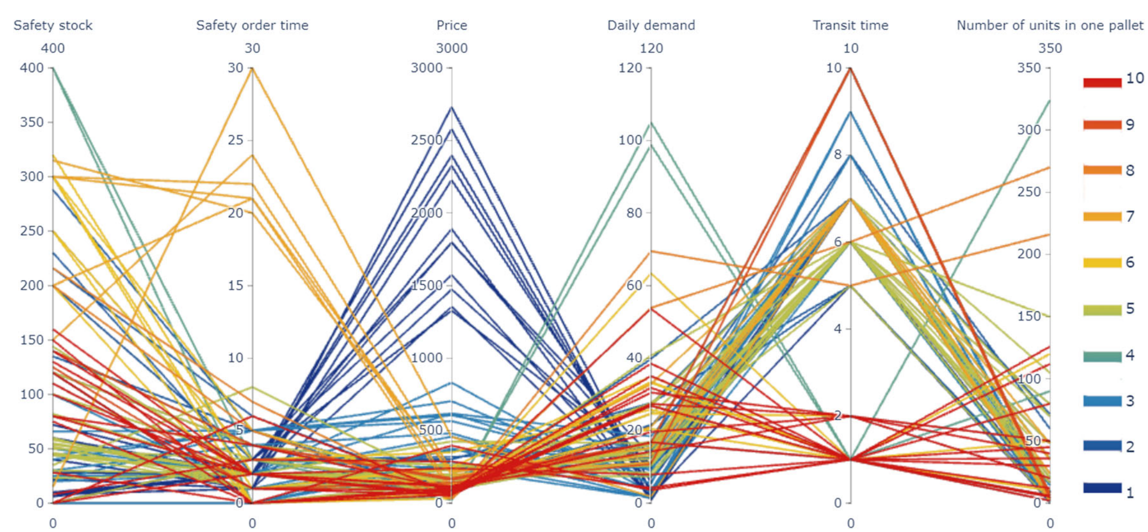

Fig. 3 Result of clustering (ten clusters indicates by different colours)

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

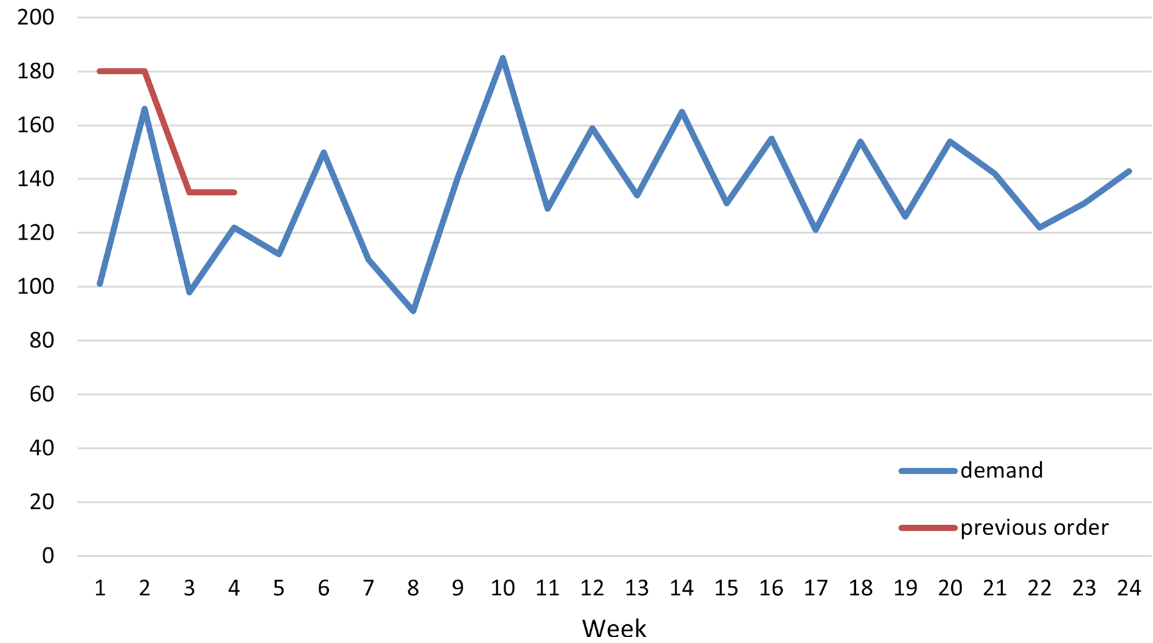

Fig. 4 Demand and previous order data for the cluster center

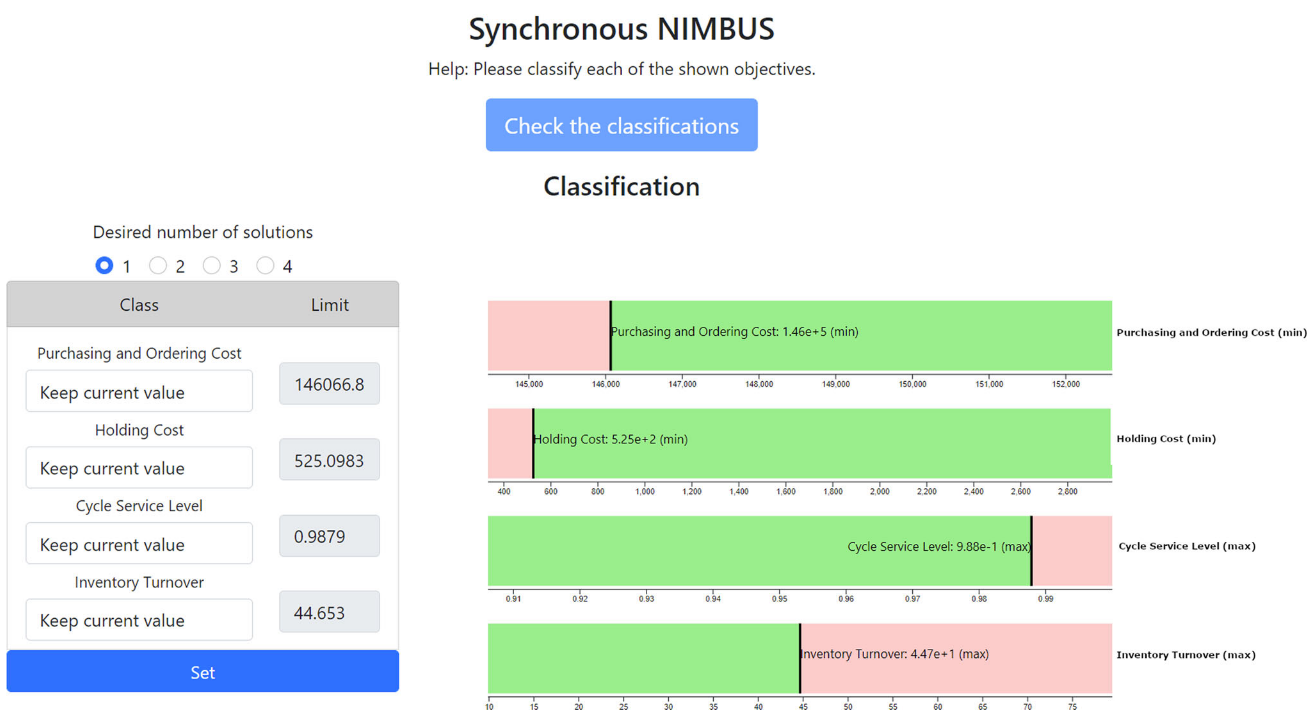

Fig. 5 Graphical user interface of NIMBUS in DESDEO

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,

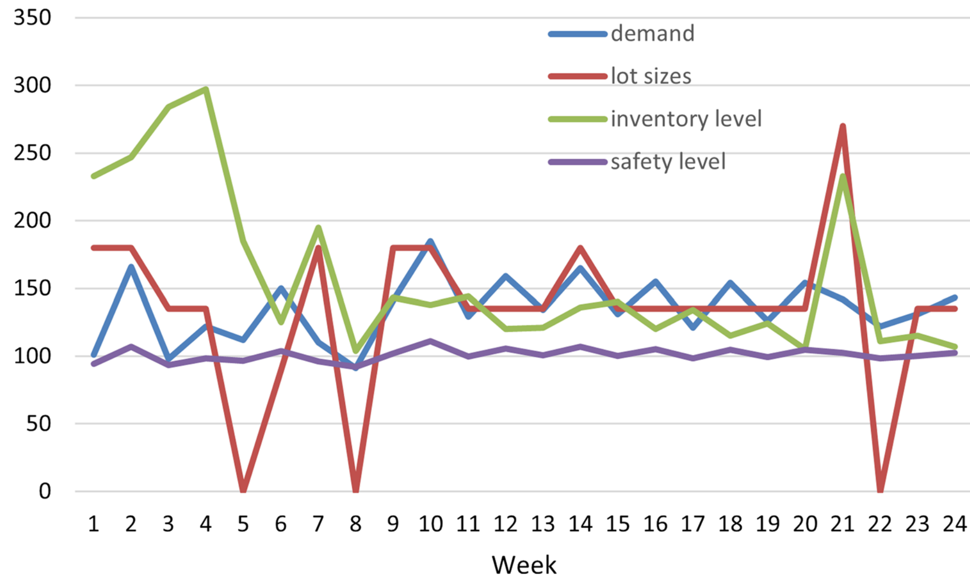

Fig. 6 Result for cluster center

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 $\boldsymbol{zs} = (146\,066.8, 525.1, 0.98, 44.65)$. With these points, we calculated the reference direction of the cluster center as $\boldsymbol{zr} = (3\,769.05, -32.11, 0.012132, -3.53)$ and the normalization of $\boldsymbol{zr}$ was $\dot{\boldsymbol{zr}} = (0.0258, -0.0612, 0.01228, -0.07896)$.

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.