Decision Analytics Journal 7 (2023) 100210

Contents lists available at ScienceDirect

Decision Analytics Journal

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

Transfer-stable aggregation functions: Applications, challenges, and emerging trends

Zbyněk Kurač

Tomas Bata University in Zlín, Faculty of Applied Informatics, The Department of Mathematics, Nad Stráněmi 4511, 760 05, Zlín, Czech Republic

ARTICLE INFO

MSC:
06A05
06B35
08A70
39B52
94A11

Keywords:
Transfer-stability
Aggregation function
Finite lattice
Transfer principle
Horizontal sums
Distance-stable lattice

ABSTRACT

The original transfer-stable aggregation functions generalized the arithmetic means to finite chains. The idea of applying these functions was later demonstrated by purchasing several products depending on the quality and price of the products. This paper aims to continue this idea and show other possible applications of transfer-stable aggregation functions. We identify several concerns in various applications and present possible remedies to address these concerns. We show different types of lattices could be used to compile the assignment of a given application problem. Based on this finding, we can very effectively divide the products into so-called qualitative classes. We conclude that distance-stable lattices are most effective in these applications. Moreover, we also show that the classes better reflect reality using these lattices.

Contents

1. Introduction

The theory of aggregation functions has deep roots dating back to the beginnings of the use of the arithmetic mean [1–4] or also to the first use of the collected data from which a conclusion was drawn in the form of a single value. Aggregation is a process where a result is derived on the basis of some data (primarily numerical) to the form of a single value. The mathematical theory dealing with this process is called the theory of aggregation functions [5–11]. Aggregation functions mathematically realize the result as a single value and satisfy two natural conditions: the aggregation function is non-decreasing and satisfies the boundary conditions. It should be noted that these conditions are not universal. In the entire history of the study of the aggregation process, it is possible to find more general functions that the result of these functions is also a certain type of a single value. However, a general study of the aggregation process has shown that most of the functions used to process statistical data meet the two mentioned conditions.

For a long time, aggregation functions were used only on real numbers, respective number sets in general, due to statistics as a

E-mail address: Kury.Z@seznam.cz.

https://doi.org/10.1016/j.dajour.2023.100210
Received 16 February 2023; Received in revised form 14 March 2023; Accepted 24 March 2023
Available online 28 March 2023

numerical discipline. Recently, however, other structures have become more prominent, especially non-linear ones, where the theory of aggregation functions manifests differently in some cases [12–15]. One such group of non-linear structures is bounded lattices [16–24] (a partially ordered set, where for every two elements there is a supremum and an infimum), where the above two conditions for the aggregation function A on the bounded lattice L are in the form: for $\mathbf{x} = (x_1, x_2, \dots, x_k), \mathbf{y} = (y_1, y_2, \dots, y_k) \in L^k$ we have

(1) $\mathbf{x} \leq \mathbf{y} \Rightarrow A(\mathbf{x}) \leq A(\mathbf{y})$,
(2) $A(0, \dots, 0) = 0$ and $A(1, \dots, 1) = 1$.

The arithmetic mean is probably the best known aggregation function, and its form is known on real numbers. However, our focus on finite bounded lattices raises a question. How can arithmetic mean be modeled in these structures? In the paper [25], it was shown that the arithmetic mean could not be defined on nonlinearly ordered structures, but it can be converted from infinite chains (real numbers) to finite chains. This transformation created so-called transfer-stable means. These functions are equivalent to the arithmetic mean on finite chains, as shown in [26]. Also, these functions can only be defined on chains because of idempotence. In the paper [26], idempotence was removed, and the so-called transfer-stable aggregation function was created. These functions can already be defined on any finite lattice. Despite this new fact, it turned out that these functions are easiest to define on the so-called transfer-stable lattices.

This paper is a continuation of the paper [26], especially the last section on the appropriate application of these functions. It has been shown that transfer-stable aggregation functions are closely related to the business strategy of purchasing [27–32] more products for the lowest cost and best quality. This was also the motivation for writing this paper to show the strength of transfer-stable aggregation functions in the business sector and their possible use in practice. Further motivation can be to choose the right combination of products when buying several products at once or choosing one product depending on several parameters. Moreover, in the final examples, we show how the theory of lattices can be effectively applied in real life [33,34].

Transfer-stable aggregation functions guide us in choosing precisely the products we want and need from a large number of products. It is clear that there are many ways to achieve this aim, and this paper describes one of them. Some might say that the use of this method is very limited in practice, but our goal is not to cover the entire business strategy for multi-product purchases. We just want to outline some possible scenarios for using this method.

The main idea of this business strategy is to divide products into lattices (usually, products of the same type into one lattice), which are then combined (using the direct product) into one big lattice (lattice of all products).[1] By applying a transfer-stable function to this lattice, we get several possible classes (qualitative classes). Each qualitative class contains equivalent purchases, which means that two selections from the same qualitative class have the same price/quality ratio.

The first problem is to find the appropriate (the best) class, the so-called golden mean. Generally, this is the class that fits the buyer and the seller (they agree that this class is a good option for both sides). In other words, this is the class from which we want to choose our final purchase. Therefore, the second problem is the choice of our final purchase. Moreover, we will try to add different priorities to the business strategy that will help to select the golden mean and the final purchase in this class.

The main aim of this paper is to demonstrate several real-life examples to show that transfer-stable aggregation functions can also be extended to the area of multicriteria decision-making [35–43]. In addition, for each example, the goal will be to identify the golden mean (the properly chosen qualitative class) and to select the right final choice from this class based on the given conditions for purchasing several products.

For a better understanding, we note (for a better understanding of the paper) that each combination of products or parameters will be written in angle brackets, i.e.,

$$\Big\langle \text{product } 1, \text{product } 2, \text{product } 3, \text{product } 4, \text{product } 5 \Big\rangle$$

or

$$\Big\langle \text{parameter } A, \text{parameter } B, \text{parameter } C, \text{parameter } D, \text{parameter } E \Big\rangle,$$

where each product or parameter has its own lattice (same or different).

To make the given examples realistically accurate, the default data for each example was taken from the website: http://alza.cz/.

2. Transfer-stable preliminaries

At the beginning we will recall important terms from the theory of transfer-stable aggregation functions. The first and the most important term is the so-called transfer-stability, which is closely related to the distance in the lattice according to the corresponding Hasse diagram.

Definition 2.1. Let $(L, \leq)$ be a finite lattice and $x, y \in L$. Then y *covers* x (y is a *successor* of x), written $x \prec y$, if $x < y$ and there is no $z \in L$ such that $x < z < y$. Moreover, the notation $x \preceq y$ means $x \prec y$ or $x = y$.

Definition 2.2. A function $A : L^k \to L$ on the finite lattice L is called *transfer-stable*, if

$$A(x_1, \dots, x_i, \dots, x_j, \dots, x_k) = A(x_1, \dots, y_i, \dots, y_j, \dots, x_k)$$

for any $i, j \in \{1, \dots, k\}$ and $x_i, y_j \in L$ where $x_i \prec y_i$ and $y_j \prec x_j$.

The class of all transfer-stable aggregation functions on the finite lattice L is denoted by the symbol TSAgg_L and the symbol $\mathsf{TSAgg}_L^{(k)}$ denotes all k-ary transfer-stable aggregation functions on the lattice L.

In other words, if we reduce one element of the input values $x_1, \dots, x_k$ down and another element up, then the result of the function does not change. For example, for the arithmetic mean AM we have: $AM(5, 7) = AM(4, 8) = AM(3, 9) = 6$.

Definition 2.3. Let L be a finite lattice. Then *distance* between elements $a, b \in L$, $a \leq b$ is the number of edges (line segments) in Hasse diagram between a and b in the lattice L.

The symbol $\mathcal{D}_{ab}$ denotes the set of all distances between the elements a and b. Denote $d_L(a, b) := \min \mathcal{D}_{ab}$ the least distance between a and b and $D_L(a, b) := \max \mathcal{D}_{ab}$ the greatest distance between a and b.

Moreover, *the path* from a to b in the lattice L is the sequence of elements $x_1, \dots, x_n \in L$ such that

$$a = x_1 \prec x_2 \prec \dots \prec x_{n-1} \prec x_n = b.$$

Whenever $P = \{a = x_1, x_2, \dots, x_{n-1}, x_n = b\}$ is a path from a to b then the symbol $|P|$ denotes the number of elements in the path P, that means $|P| = n$ and the symbol $\|P\|$ denotes the path length (the number of edges in the path P) from a to b in the lattice L, that means $\|P\| = |P| - 1$. The most important paths in the lattice L are paths from 0 to 1 and all these paths are denoted by the symbol $\mathrm{Path}L$.

Based on the definition of distance in the lattice, the class of all lattices was divided into two disjoint subclasses with respect to the unambiguous distance in the lattice. Thus we get the definition of the so-called distance-stable lattice.

Definition 2.4. The finite lattice L is called *distance-stable* if the distance between each two elements is unambiguously defined, i.e., $d_L(a, b) = D_L(a, b)$ for all $a, b \in L$, $a \leq b$.

[1] Please pay attention to the difference in the word product. The direct product is the operation, but the product is the good in the store.

Obviously, if the lattice L is distance-stable, then $d_L(0,1) = D_L(0,1)$. This value is called *depth of the lattice*. Moreover, we can omit the index L if it is known which lattice it is.

The main characteristics of transfer-stable aggregation functions are the so-called (transfer-stable) blocks, which are individual layers of the transfer-stable aggregation function and these layers have the same final value for different input values.

Definition 2.5. The subset of k-tuples of L^k, which are linked together by transfer-stability, is called a *(transfer-stable) block*, that is, two elements $\mathbf{x}, \mathbf{y} \in L^k$ belong to the same block if $TS(\mathbf{x}) = TS(\mathbf{y})$ for all $TS \in \mathsf{TSAgg}_L^{(k)}$.

In other words, the elements $\mathbf{x}, \mathbf{y} \in L^k$ belong to the same block if their distances from the smallest element $\mathbf{0}$ of the lattice L^k are the same, i.e., $d_{L^k}(\mathbf{0}, \mathbf{x}) = d_{L^k}(\mathbf{0}, \mathbf{y})$, denote by $\mathbf{x} \in [\mathbf{y}]$. We indicate the block in to square brackets }}[•]ε because it is an equivalence class. For this reason, we will call the blocks a class (later, qualitative class). It should be noted that each class is determined by its arbitrary element, i.e., the class $[\mathbf{x}]$ represented by element $\mathbf{x}$ is the same as the class $[\mathbf{y}]$ represented by element $\mathbf{y}$, because $\mathbf{x} \in [\mathbf{y}]$.

Next, if we apply the k-ary aggregation function to the distance-stable lattice L, then we can determine the number of all (blocks) classes very easily (that is, the number of all blocks in the lattice L^k).

Proposition 2.6. *Let L be a distance-stable lattice. Then the number of all blocks in the distance-stable lattice L^k is equal to $k \cdot d_L(0,1) + 1$.*

In the paper [26] it was shown that blocks can be ordered in two ways: linear or zig-zag (the blocks alternate with each other). The first type is more important and thus the so-called transfer-stable lattices were created.

Definition 2.7. The finite lattice L is called *transfer-stable*, if (transfer-stable) blocks in the lattice L^2 (and therefore every other power) are linearly ordered.

It is important to identify transfer-stable and transfer-unstable lattices. The following statement characterizes transfer-stable lattices.

Theorem 2.8. *The finite lattice L is a transfer-stable if and only if it is a distance-stable or there are paths $P_1, P_2, P_3 \in \mathrm{Path}L$ such that*

$$\gcd\left(\|P_1\| - \|P_2\|, \|P_1\| - \|P_3\|, \|P_2\| - \|P_3\|\right) = 1,$$

where the symbol gcd *means the greatest common divisor of elements.*

We need to mention one important fact about Theorem 2.8. This is a note on the assumption, because we have to take into account that the assumption $P_1 \neq P_2 \neq P_3$ is not used. That means, for example, lattices contain no more than two paths from 0 to 1, then two paths can be taken the same.

According to Theorem 2.8, the so-called horizontal sums [44–46] are typical lattices, where it is easy to identify whether they are a transfer-stable lattices or not.

Definition 2.9. Let $n \in \mathbb{N}$ a $L_1, \dots, L_n$ be a boundary chains with 0 and 1. Then lattice L is called *horizontal sum* of lattices $L_1, \dots, L_n$, if

(i) $L = \bigcup_{i=1}^{n} L_i$,
(ii) $L_i \cap L_j = \{0,1\}$ for all $i,j \in I$, $i \neq j$,
(iii) $a \vee b = 1$ and $a \wedge b = 0$ for all $a \in L_i \setminus \{0,1\}$, $b \in L_j \setminus \{0,1\}$, where $i,j \in I$, $i \neq j$.

In other words, the lattice L is a horizontal sum if and only if for every two incomparable elements $a, b \in L$ holds that their join is 1 and meet is 0, i.e., $a \vee b = 1$ and $a \wedge b = 0$.

The last term to be recalled is the so-called first unstable element relative to 0 and 1, which plays an important role in determining all classes of transfer-stable aggregation function in a transfer-stable lattice.

Definition 2.10. The element α is called *the first unstable element relative to* 0, if the set $\mathcal{D}_{0\alpha}$ is not the single-element set and the set $\mathcal{D}_{0\gamma}$ is the single-element set for all $\gamma \in L$ such that $d(0,\gamma) < d(0,\alpha)$. Similarly, the element β is called *the first unstable element relative to* 1 if the set $\mathcal{D}_{\beta 1}$ is not the single-element set and the set $\mathcal{D}_{\gamma 1}$ is the single-element set for all $\gamma \in L$ such that $d(\gamma,1) < d(\beta,1)$.

Based on the first unstable element relative to 0 and 1, we can pronounce a theorem on the number of all blocks for transfer-stable distance-unstable lattices. In essence, the number of classes in the direct product of these lattices does not change with the increasing power of the direct product.

Proposition 2.11. *The number of all blocks of transfer-stable distance-unstable lattice L is the same for all k-ary transfer-stable aggregation functions of the lattice L, that means $d(0,\alpha) + d(1,\beta) + 1$.*

A more detailed description of all the mentioned properties can be found in the papers [25,26].

3. Introductory example

Let us start with a very simple example. A similar example can be found at the end of the paper [26]. To better understanding all the terms introduced, we recommend reading the last example in the paper [26].

Example 3.1. Let us imagine, we are a team leader in a company and our task is to buy new mobile phones for our team members. The team has six employees $Z_1, \dots, Z_6$, so we have to buy six mobile phones. Therefore, we use the 6-ary transfer-stable aggregation function. In contrast to the example from the paper [26], this time we will use two distance-stable lattices L_1 and L_2, where the first lattice represents the experience of workers and the second one represents the quality of mobile phones.

The lattice L_1 (see Fig. 1) is created by a direct product of two three-element linguistic chains, where the first one has elements

$$\{\text{New}, \text{Short-term}, \text{Long-term}\}$$

team worker (abbreviated as N, S, L) and the second one contains the experience of the workers (abbreviated as T, H, E), i.e.,

$$\{\text{Tolerable}, \text{Hardworking}, \text{Expert}\}.$$

In this case ($k = 6$), we get 25 classes[2] representing the strength of the team. That is, the team containing six new employees, who have not shown anything for the team yet, will have a strength equal to **0**. On the other hand, a team of long-time experts has a strength equal to **24**. Our team consists two new employees, where one of them is tolerable and the other one is hardworking, three short-term employees (tolerable, hardworking, expert) and one long-term expert, i.e., $\langle NT, NH, ST, SH, SE, LE \rangle$, and the team strength is equal to **11**, because[3]

$$\langle NT, NH, ST, SH, SE, LE \rangle \in \left[\langle ST, LT, LT, LT, LT \rangle\right].$$

[4] Since the average class is **12** ($\frac{0+24}{2} = 12$), we could say that our team is quite average.

[2] $6 \cdot d_{L_1}(0,1) + 1 = 6 \cdot 4 + 1 = 25$

[3] $d(NT,NT) = 0$, $d(NT,NH) = 1$, $d(NT,ST) = 1$, $d(NT,SH) = 2$, $d(NT,SE) = 3$, $d(NT,LE) = 4$, then $0+1+1+2+3+4 = 11$.

[4] Qualitative class **11** is represented by the 5-tuple $\langle ST, LT, LT, LT, LT \rangle$, where ST is Short-term Tolerable and LT is Long-term Tolerable.

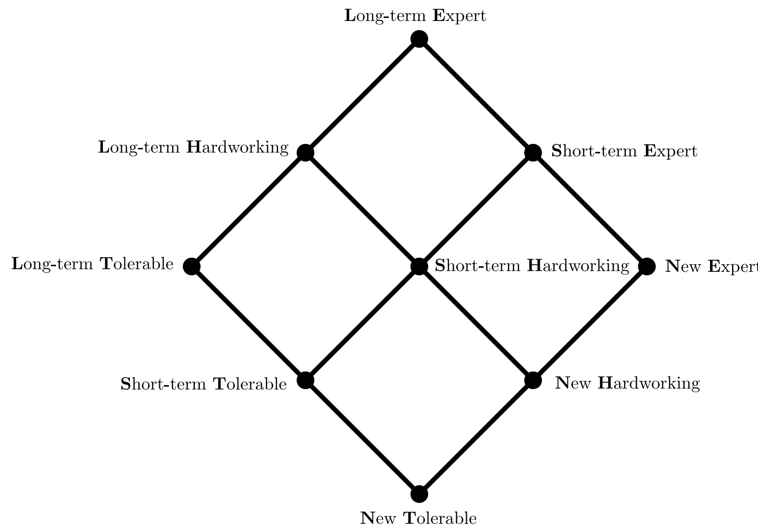

Fig. 1. Distance-stable lattice L_1 representing the employee's experience and time spent in the team.

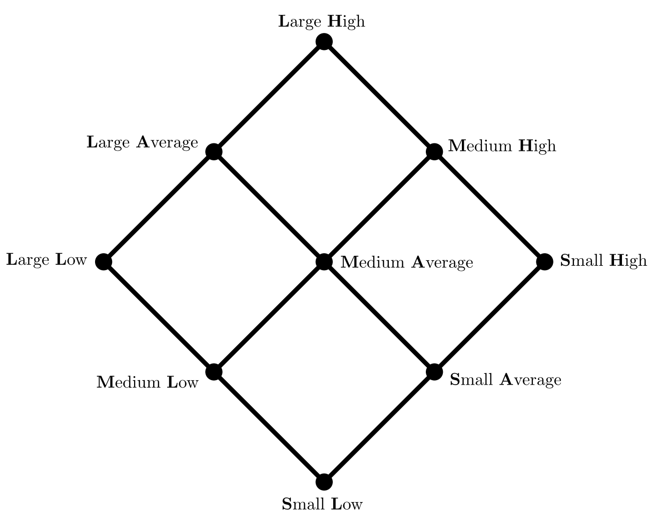

Fig. 2. Distance-stable lattice L_2 representing the display size and performance of the mobile phone.

Now, the lattice L_2 (see Fig. 2) is also the direct product of two three-element linguistic chains representing the display size of the mobile phone and the mobile performance, respectively, i.e.,

$\{\text{Small}, \text{Medium}, \text{Large}\}$

display size (abbreviated as S, M, L) and

$\{\text{Low}, \text{Average}, \text{High}\}$

performance of the mobile phone (abbreviated as L, A, H).

Again, we get 25 classes for six employees. Unfortunately, the highest class is not the best here, because the display size of the mobile phone cannot be compared from the best to the worst. On the other hand, the performance of the mobile phone can be ordered from the worst to the best. That means, classes[5]

$$[\langle SH, SH, SH, SH, SH, SH\rangle] \ \dots \ \mathbf{12}$$
$$[\langle MH, MH, MH, MH, MH, MH\rangle] \ \dots \ \mathbf{18}$$
$$[\langle LH, LH, LH, LH, LH, LH\rangle] \ \dots \ \mathbf{24}$$

could be considered the best in some respects.

[5] Each class contains High-performance mobile phones with different display sizes.

In order to achieve the correct result in this situation, it is necessary to include another variable in the calculations—priorities. Each employee has different priorities about display screen size. Suppose an employee Z_2 likes a small display, employees Z_1, Z_3 and Z_5 want a medium display, and employees Z_4 and Z_6 require a large display of the mobile phone. Based on these requirements, we can declare which class is the worst and which is the best in this case. If we hear the wishes of our employees, either we can get a class for low performance mobile phones

$$[\langle SL, ML, ML, ML, LL, LL\rangle] \ \dots \ \mathbf{7},$$

or we can get a class for high performance mobile phones

$$[\langle SH, MH, MH, MH, LH, LH\rangle] \ \dots \ \mathbf{19}.$$

If we combine this fact with the fact that our team is almost average, we can find the average class from the above mentioned (between classes **7** and **19**). The result is the class **13**, and so for our team we choose the class **12** as a golden mean, which is characterized by 6-tuple

$$\langle LL, LL, LL, LL, LL, LL\rangle.$$

Based on transfer-stability, we have the following 6-tuples in the class **12**:

$$\langle SL, ML, ML, MA, LH, LH\rangle \ ; \ \langle SA, ML, ML, ML, LH, LH\rangle$$
$$\langle SL, MA, MA, MA, LA, LA\rangle \ ; \ \langle SA, ML, MA, MA, LA, LA\rangle$$
$$\langle SH, ML, ML, MA, LA, LA\rangle \ ; \ \langle SL, MA, MH, MH, LL, LL\rangle$$
$$\langle SA, MA, MA, MH, LL, LL\rangle \ ; \ \langle SH, MA, MA, MA, LL, LL\rangle$$
$$\langle SL, MA, MA, MA, LL, LH\rangle \ ; \ \langle SA, MA, MA, MA, LL, LA\rangle$$
$$\langle SL, ML, ML, MH, LA, LH\rangle \ ; \ \langle SH, ML, MA, MA, LL, LA\rangle$$
$$\langle SL, ML, MA, MH, LL, LH\rangle \ ; \ \langle SA, ML, MA, MH, LL, LA\rangle$$
$$\langle SH, ML, MA, MH, LL, LL\rangle \ ; \ \langle SL, ML, MA, MH, LL, LH\rangle$$
$$\langle SH, MH, ML, ML, LL, LA\rangle \ ; \ \langle SL, ML, MA, MA, LA, LH\rangle$$

It is only our decision what combination we will use for our team members. We can take advantage of the above-mentioned fact that employees Z_1 and Z_2 are new, but Z_1 is tolerable and Z_2 is hardworking. Employees Z_3, Z_4 and Z_5 are in the team for a short time, but he or she is a tolerant, hardworking and expert employee, respectively. The last employee is an employee Z_6, who is a long-term expert in the team. In addition, we define the map

$$(x_1, x_2, x_3, x_4, x_5, x_6) \mapsto (Z_2, Z_1, Z_3, Z_5, Z_4, Z_6)$$

for all $x_1, x_2, x_3, x_4, x_5, x_6 \in L_2$. That means, for instance, 6-tuple

$$\langle SH, MA, MA, MA, LL, LL \rangle$$

can be described as follows: the employee Z_2 receives a small-display, high-performance mobile phone, employees Z_1, Z_3 and Z_5 receive a mobile phones with a medium display and average performance and employees Z_4 and Z_6 get a low-performance mobile phone with a large display.

On the other hand, for example, the 6-tuple of mobile phones $\langle SH, MH, ML, ML, LL, LA \rangle$ can be a situation, when we give powerful mobile phones to new employees to keep them in the team, short-term employees receive low-performing mobile phones and we have to give an mobile phone with average performance to a long-term expert based on the performance of the first two mobile phones. Conversely, according to the second 6-tuple from the last line of the list for class **12**, i.e., $\langle SL, ML, MA, MA, LA, LH \rangle$, we buy a high-performance mobile phone for a long-term expert, average-performance mobile phone is enough for short-term employees and new employees have to show their potential, and therefore they receive low-performance mobile phones only.

However, our final decision will be the 6-tuple from the sixth line of the list above, i.e.,

$$\langle SL, ML, ML, MH, LA, LH \rangle,$$

because we will buy two high-performance mobile phones to our experts. For the short-term hardworking employee, an average-performance mobile phone is enough for now and as for the others, either they are brand new employees or those who have not yet shown their full potential and therefore they will receive low-performance mobile phones.

It should be noted that we could also select the appropriate qualitative class (golden mean) on a budget basis. Considering that the price of a low-performance mobile phone is around 5 000 CZK, average-performance is around 15 000 CZK and the price of a high-performance mobile phone is around 25 000 CZK, then we can declare on the basis of these data that class **7**

$$[\langle SL, ML, ML, ML, LL, LL \rangle],$$

has a budget of around 30 000 CZK and the class **19**

$$[\langle SH, MH, MH, MH, LH, LH \rangle],$$

has a budget of around 150 000 CZK. However, budget allocation would not be possible without assuming the display size, because, for instance, the 6-tuple $[\langle LL, LA, LA, LA, LH, LH \rangle]$ belonging to the class **19** has a budget of about 100 000 CZK, not the required 150 000 CZK. Fortunately, assuming the size of the display of the mobile phone, we conclude that all corresponding 6-tuples (from the same qualitative class) have the same budget, as we can see in the list for the class **12**, where the budget of each 6-tuple is 80 000 CZK. Therefore, we know for sure that any 6-tuple we choose from the class **12** we will always pay around 80 000 CZK for mobile phones.

4. Sold out products as unavailable combinations

Returning to the example from the introduction and example from paper [26], we can notice the unspoken assumption that there must exist all combinations of the products shown in the lattice of all products. Nevertheless, in general, it may happen that there is no product belonging to some combination of price and quality. In this case, we would have to exclude this element (combination) from our considerations and also remove the corresponding element from the lattice of all products. The problem would arise if we had to remove more such elements but all new products would no longer form a lattice. Under these circumstances, we would not be able to apply the theory of transfer-stable functions. Fortunately, there is a way to resolve this problem.

Table 1
Laptop parameters in the Example 4.1.

Price	Weight	Graphics card	RAM Memory
up to 10 000 CZK	up to 1 kg	Game playing	64 GB
10 000–20 000 CZK	1–2 kg	Dedicated	32 GB
20 000–30 000 CZK	2–3 kg	Integrated	24 GB
over 30 000 CZK	over 3 kg		16 GB
			12 GB
			8 GB
			4 GB

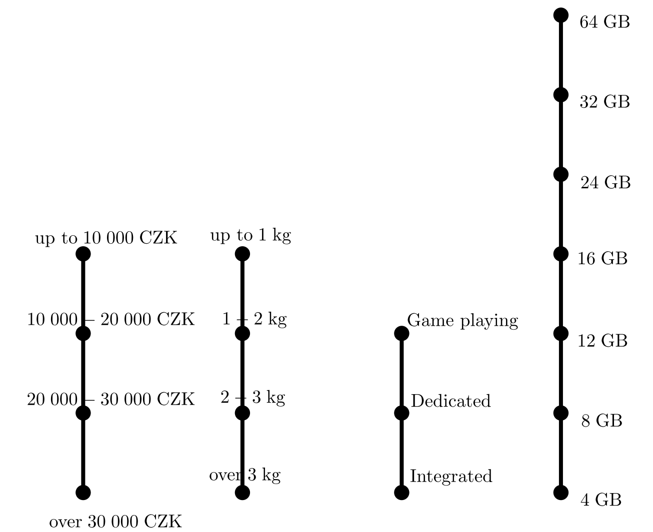

Fig. 3. Linguistic chains representing the price of the laptop, the weight of the laptop, the type of graphics card and the size of operational RAM.

We understand discarded combinations as sold-out goods. In other words, it is a combination of products that cannot be bought or a combination of parameters of a non-existing product. Still, we count on them, but if such a combination (product) appears in the option we have chosen, then we must ignore this option, as it is not complete (exist). Then we need to choose undiscarded combination (option) in the same qualitative class. If there is no such option, we must choose a new golden mean (qualitative class).

Example 4.1. The goal of this example is to buy three laptops depending on the parameters from Table 1.

As in the example from the introduction, each column of Table 1 represents one linguistic lattice (chain), i.e., the lattice representing the price of the laptop, the weight of the laptop, the type of the graphics card of the laptop and the size of operational RAM of the laptop, respectively (see Fig. 3).

Now we create a direct product of these four linguistic chains and obtain the lattice L_3, which has $4 \cdot 4 \cdot 3 \cdot 7 = 336$ elements. Since it was created as a direct product of distance-stable lattices, it is also a distance-stable lattice with depth $d(0,1) = 3 + 3 + 2 + 6 = 14$.[6] Thus, the purchase of 3 laptops indicates the use of the 3-ary transfer-stable aggregation function and we get $3 \cdot d(0,1) + 1 = 3 \cdot 14 + 1 = 43$ blocks (qualitative classes) in the lattice $(L_3)^3$. For example, three products of type[7]

$$\langle \text{over 30 000 CZK}, \text{over 3 kg}, \text{Integrated}, \text{4 GB} \rangle$$

[6] The depth of the direct product of the distance-stable lattices is equal to the sum of the depths of the distance-stable lattices.

[7] Note that in the introductory example, one combination corresponded to the final selection (purchase) because each product was easy to write down. This time, however, one product is determined by parameters, i.e. one combination of parameters (also written in angle brackets) corresponds to one product (not three, as requested). If we write these three combinations side

Table 2
The list of possible products in Example 4.1.

	♯	Price	Weight	Graphics card	RAM Memory
P_1	3	over 30 000 CZK	over 3 kg	Game playing	8 GB
P_2	3	10 000–20 000 CZK	2–3 kg	Integrated	4 GB
P_3	4	20 000–30 000 CZK	2–3 kg	Dedicated	8 GB
P_4	4	10 000–20 000 CZK	1–2 kg	Integrated	4 GB
P_5	5	over 30 000 CZK	2–3 kg	Dedicated	16 GB
P_6	5	up to 10 000 CZK	1–2 kg	Integrated	4 GB
P_7	6	up to 10 000 CZK	up to 1 kg	Integrated	4 GB
P_8	6	10 000 – 20 000 CZK	1–2 kg	Dedicated	8 GB
P_9	7	over 30 000 CZK	1–2 kg	Game playing	16 GB
P_{10}	7	20 000–30 000 CZK	up to 1 kg	Integrated	16 GB
P_{11}	8	20 000–30 000 CZK	1–2 kg	Game playing	16 GB
P_{12}	8	over 30 000 CZK	up to 1 kg	Integrated	32 GB
P_{13}	9	over 30 000 CZK	2–3 kg	Game playing	64 GB
P_{14}	9	10 000–20 000 CZK	1–2 kg	Game playing	16 GB

belong to the qualitative class **0**, while three products of type

$$\langle \text{up to } 10\,000 \text{ CZK}, \text{up to } 1 \text{ kg}, \text{Game playing}, 64 \text{ GB} \rangle$$

correspond to the qualitative class **42**.

It should now be noted that none of these products can be purchased in this configuration. Therefore, we consider them sold out and the qualitative classes **0** and **42** cannot be selected.

As a golden mean, it is possible to choose the qualitative class **29** and thus all three products must have a total distance from the smallest element in the lattice L_3 equal to **29** based on the theory of distance-stable lattices.

To illustrate, the product

$$\langle 10\,000 - 20\,000 \text{ CZK}, 1 - 2 \text{ kg}, \text{Dedicated}, 24 \text{ GB} \rangle$$

has a distance of 9 from the smallest element in the lattice L_3, because

- $d(\text{over } 30\,000 \text{ CZK}, 10\,000 - 20\,000 \text{ CZK}) = 2$
- $d(\text{over } 3 \text{ kg}, 1 - 2 \text{ kg}) = 2$
- $d(\text{Integrated}, \text{Dedicated}) = 1$
- $d(4 \text{ GB}, 24 \text{ GB}) = 4$,

whence in the sum we have the value $2+2+1+4 = 9$. In a similar way we find that the distances of the products

$$\langle 10\,000 - 20\,000 \text{ CZK}, 1 - 2 \text{ kg}, \text{Game playing}, 24 \text{ GB} \rangle,$$
$$\langle 10\,000 - 20\,000 \text{ CZK}, 1 - 2 \text{ kg}, \text{Dedicated}, 32 \text{ GB} \rangle$$

are equal to 10. In total, these three products have a distance of $9 + 10 + 10 = 29$ from the smallest element in the lattice $(L_3)^3$.[8] As a consequence, these three products belong to the qualitative class **29**. Unfortunately, it is not possible to buy these products even now.

Trying other options in the qualitative class **29**, we would conclude that none of the options are available. In other words, no three products belonging to the qualitative class **29** can be selected.

Further examination of this example and possible purchases, we conclude that the original parameters of laptops (see Table 1) are not well chosen. The highest available class[9] is the qualitative class **27**, because there is no available product with a distance of 10,[10] but the product

$$\langle \text{over } 30\,000 \text{ CZK}, 2 - 3 \text{ kg}, \text{Game playing}, 64 \text{ GB} \rangle$$

with the distance of $0+1+2+6 = 9$ can be bought. Similarly, we would find that the lowest available class is **9**, because there is no available product with a distance of 0,1 or 2, but the product

$$\langle 10\,000 - 20\,000 \text{ CZK}, 2 - 3 \text{ kg}, \text{Integrated}, 4 \text{ GB} \rangle$$

with the distance of $2 + 1 + 0 + 0 = 3$ can be bought. To clarify, each triplet of products belonging to the class less than **9** and more than **27** contains at least one sold out (unavailable) product.

Based on the new facts, we need to change the golden mean. The recommended new qualitative class is **15**. We will now create a list of several products ordered by distance from the smallest element (♯) in the lattice L_3.

Using transfer-stability, we can declare that the following potential purchases of three products from Table 2 are equivalent, i.e., they belong to the same qualitative class:[11]

$$\langle P_{14}, P_1, P_2\rangle; \langle P_{13}, P_1, P_1\rangle; \langle P_{12}, P_4, P_1\rangle; \langle P_{11}, P_3, P_2\rangle; \langle P_{10}, P_6, P_2\rangle;$$
$$\langle P_9, P_4, P_3\rangle; \langle P_8, P_7, P_1\rangle; \langle P_7, P_6, P_4\rangle; \langle P_6, P_5, P_5\rangle; \langle P_5, P_5, P_5\rangle.$$

In this case, it is the class **15**, because the sum of the distances of the individual products from the smallest element in the lattice L_3 is equal to 15.

The last and most important task in this example is choosing our purchase. The choice from the above mentioned options depends on the given preferences. In particular, we can select the final purchase according to:

- price: $\langle P_7, P_6, P_4\rangle$
  - the price should not exceed 40 000 CZK,
- weight: $\langle P_7, P_6, P_4\rangle$
  - the weight should not exceed 5 kg,
- type of graphics card: $\langle P_{13}, P_1, P_1\rangle$
  - three game playing graphics cards,
- size of operational RAM: $\langle P_{13}, P_1, P_1\rangle$
  - the total size of the operational RAM is 80 GB.

5. Incomparable parameters

In the example 3.1 we could notice a minor problem with certain parameters. Specifically, it is about incomparability of parameters. For the display size parameter, we cannot say which size is better or worse.

by side, then we get the same as in the introductory example. Instead, we just say (to make the notation not too complicated) that we have three such combinations.

[8] Note that one product belongs to lattice L_3, but three products belong to lattice $(L_3)^3$. This means that the distance of one product is measured in the lattice L_3, but the total distance (of all three products) is considered in the lattice $(L_3)^3$.

[9] This is a class where there is at least one combination of parameters (product) that can be purchased. A product with this combination of parameters exists.

[10] If such a product existed, then a trio of such products would belong to class **30**, which would mean that this class would be available.

[11] This notation is the same as in the introductory example.

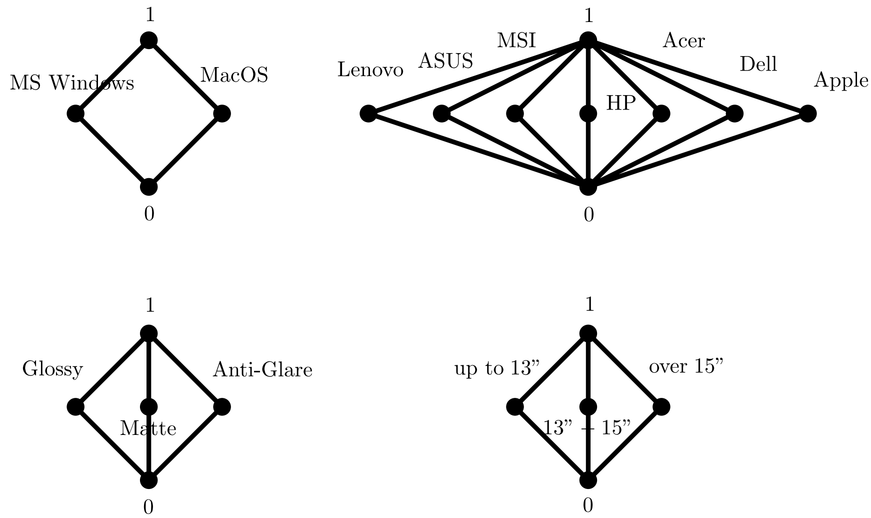

Fig. 4. Linguistic n-dimensional ($n = 2, 7, 3, 3$) diamonds representing, the operating system of the laptop, the brand of the laptop, the display type and size of laptop screen, respectively.

Table 3
Laptop parameters in the Example 5.1.

Operating system	Brand	Display type	Display size
Microsoft Windows	Lenovo	Glossy	up to 13"
MacOS	ASUS	Matte	13"–15"
	MSI	Anti-Glare	over 15"
	HP		
	Acer		
	Dell		
	Apple		

Nevertheless, we ordered the sizes of display and created a linguistic chain. Now, we look first at the examples that have at least one incomparable parameter in the task and show the solution for these parameters. Based on the presented procedure in these examples, we will return to the example 3.1 and solve it from a different point of view.

Example 5.1. As in the example 4.1, the goal is to buy three laptops, but this time according to different parameters. The new parameters are presented in Table 3.

Using the same principle as in the previous example, we obtain four linguistic chains representing the operating system of the laptop, the brand of the laptop, the display type and size of the laptop, respectively. By their direct product we get a distance-stable lattice L_4 with 126 elements and a depth of 11. With the purchase of three products, $3 \cdot 11 + 1 = 34$ classes are created.

Subsequently, we could declare that three products of type

$$\langle \text{MacOS, Apple, Anti-Glare, over 15"} \rangle$$

belong to the class **0** and three products of type

$$\langle \text{Microsoft Windows, Lenovo, Glossy, up to 13"} \rangle$$

belong to the class **33**. However, it is not true that the first purchase is the worst and the second is the best, so this approach is incorrect and we have to move to a different form of linguistic lattices. We use the so-called n-dimensional diamonds[12] instead of chains. The elements 0 and 1 represent unavailable parameters, that means, any product will automatically be considered sold out if it contains these (unavailable) parameters.

[12] The lattice with the smallest and the largest element and all its other elements (except the smallest and largest one) are incomparable.

Again, we get four distance-stable lattices (see Fig. 4), so the resulting lattice L_5 will also be distance-stable lattice. The direct product L_5 has $4 \cdot 9 \cdot 5 \cdot 5 = 900$ elements with depth of $2 + 2 + 2 + 2 = 8$, so we get $3 \cdot 8 + 1 = 25$ classes.

However, the only reasonable qualitative class is **12** because we can only select the three available products in this class and make our planned purchase. For example, either three products with parameters

$$\langle \text{ MS Windows, Lenovo, Glossy, up to 13"} \rangle$$

or another three products with parameters

$$\langle \text{ MacOS, Apple, Anti-Glare, over 15"} \rangle$$

belong to the class **12**, because the distance of any applicable parameter from the smallest element in any n-dimensional diamond is 1, i.e., $4+4+4 = 12$. In any higher or lower class then **12**, there is always a product that contains the parameter 0 or 1, so it is not possible to buy it. Such as products

$$\langle 0, \text{MSI}, 1, \text{13" – 15"} \rangle \ ; \ \langle \text{MS Windows}, 1, \text{Glossy}, 0 \rangle \ ; \ \langle \text{MacOS}, 0, 1, 1 \rangle$$

belong to the class **13**. Note that a product with an unavailable parameter may also exist in the class **12**. As proof, products

$$\langle 0, \text{MSI}, 1, \text{13" – 15"} \rangle \ ; \ \langle \text{MS Windows}, 1, \text{Glossy}, 0 \rangle \ ; \ \langle 0, 0, 1, 1 \rangle$$

belong to the class **12**, but none of the products can be bought. We see that the only possible class is the class **12**, so there is no consultation on the golden mean, because there is exactly one golden mean in this case.

If we compare this conclusion with the first half of this example (where we considered four linguistic chains), this time we have a real scenario, because all laptops with parameters from the Table 3 are equivalent (in the same qualitative class) and we cannot say which is better or worse. Therefore, the final purchase depends on our requirements and the availability of the required products. To illustrate, it is not possible to buy the products

$$\langle \text{MS Windows, MSI, Anti-Glare, up to 13"} \rangle$$
$$\langle \text{MacOS, HP, Glossy, 13" – 15"} \rangle$$
$$\langle \text{MacOS, Apple, Matte, over 15"} \rangle$$

because all three products are sold out (a laptop in this combination cannot be bought). It is also not possible to purchase products

$$\langle 0, 1, 0, 1 \rangle$$
$$\langle \text{MS Windows, Acer, Glossy, 13" – 15"} \rangle$$
$$\langle \text{MacOS}, 0, 1, \text{up to 13"} \rangle,$$

because some of the chosen products have unavailable parameters. On the other hand, the products

$$\langle \text{MS Windows}, \text{ASUS}, \text{Matte}, \text{up to } 13\text{''} \rangle$$
$$\langle \text{MS Windows}, \text{Dell}, \text{Anti-Glare}, 13\text{''} - 15\text{''} \rangle$$
$$\langle \text{MacOS}, \text{Apple}, \text{Glossy}, \text{over } 15\text{''} \rangle$$

can be purchased without any problems.

In this example, all parameters were incomparable, resulting in the existence of the only one (available) qualitative class. In the following example we will show the connection of a comparable and incomparable parameter, i.e., the direct product of the linguistic chain and the n-dimensional diamond.

Example 5.2. Let us consider the parameter "Price" from the Table 1 (linguistic four-element chain) and the parameter "Display size" from the Table 3 (linguistic three-dimensional diamond). By creating a direct product of these lattices, we obtain a distance-stable lattice L_6 with $4 \cdot 5 = 20$ elements and a depth of 5. We still want to buy three laptops, so the number of classes is $3 \cdot 5 + 1 = 16$, where the smallest class **0** is represented by the trio of products $\langle$over 30 000 CZK, 0$\rangle$ and the highest class **15** is represented by the trio of products $\langle$up to 10 000 CZK, 1$\rangle$.

First of all, we need to focus on the available classes, because as we found in the previous example, there are no three available products in the class **0** or **15**. The smallest available qualitative class is **3**, because it is represented by three existing products of the type $\langle$over 30 000 CZK, up to 13"$\rangle$. On the other hand, the highest available qualitative class is **12**, because it is represented by three existing products of the type $\langle$up to 10 000 CZK, up to 13"$\rangle$. From this fact we can already see that the original range of classes (between **0–15**) is not appropriate because there are classes that have no available product. The new range of classes is between **3–12**, for a total of 10 available classes.

Let us say, the golden mean could be the class **8**, which is determined by three products

$$2 \times \langle 10\,000 - 20\,000 \text{ CZK}, \text{ up to } 13\text{''} \rangle$$
$$\langle 20\,000 - 30\,000 \text{ CZK}, \text{ up to } 13\text{''} \rangle,$$

because

- $d\left(\text{over } 30\,000 \text{ CZK}, 10\,000 - 20\,000 \text{ CZK}\right) = 2$
- $d\left(\text{over } 30\,000 \text{ CZK}, 20\,000 - 30\,000 \text{ CZK}\right) = 1$
- $d\left(0, \text{ up to } 13\text{''}\right) = 1$,

therefore $(2+1)+(2+1)+(1+1) = 8$. Other possible triplets of products in this class are, for example:

(1)

$$\langle 10\,000 - 20\,000 \text{ CZK}, \text{ up to } 13\text{''} \rangle$$
$$\langle 10\,000 - 20\,000 \text{ CZK}, 13\text{''} - 15\text{''} \rangle$$
$$\langle 20\,000 - 30\,000 \text{ CZK}, \text{ over } 15\text{''} \rangle,$$

(2)

$$\langle \text{up to } 10\,000 \text{ CZK}, \text{ over } 15\text{''} \rangle$$
$$\langle 10\,000 - 20\,000 \text{ CZK}, \text{ over } 15\text{''} \rangle$$
$$\langle \text{over } 30\,000 \text{ CZK}, \text{ over } 15\text{''} \rangle,$$

(3)

$$\langle \text{up to } 10\,000 \text{ CZK}, \text{ over } 15\text{''} \rangle$$
$$\langle 20\,000 - 30\,000 \text{ CZK}, 13\text{''} - 15\text{''} \rangle$$
$$\langle 20\,000 - 30\,000 \text{ CZK}, 13\text{''} - 15\text{''} \rangle.$$

Table 4
Parameters of laptops in the Example 5.3.

Operating system	Brand	Weight	Graphics card
Microsoft Windows	Lenovo	up to 2 kg	Game playing
MacOS	ASUS	over 2 kg	Dedicated
	MSI		Integrated
	HP		
	Acer		
	Dell		
	Apple		

In this case, we do not have to pay attention to the combination of parameters, because all possible combinations of the price and the display size are available, that is, all products can be purchased. The only problem remains the occurrence of an unavailable parameter, such as products

$$\langle 10\,000 - 20\,000 \text{ CZK}, 1 \rangle ; \langle 20\,000 - 30\,000 \text{ CZK}, 1 \rangle ;$$
$$\langle 20\,000 - 30\,000 \text{ CZK}, 0 \rangle$$

also belong to the class **8**, but this purchase cannot be made.

In conclusion, we see that our choice in a given class does not depend on the display size of the laptop, but only on the price of the laptop, which is exactly what we expect.

Another example will be very similar to the previous one, but to a greater extent, to see the effectiveness of n-dimensional diamonds for incomparable parameters and involvement into the decision.

Example 5.3. We use the parameters we have already used in the previous examples. Specifically, it is the operating system of the laptop, the brand of the laptop, the weight of the laptop and the type of the graphics card. For clarity, we have created the Table 4.

We know that the "Operating system" and "Brand" are incomparable parameters, so they form two-dimensional and 7-dimensional linguistic diamond, respectively. On the other hand, the "Weight" and "Graphics card" are comparable parameters, so they form two-element and three-element linguistic chain, respectively. As a result (direct product) we get a distance-stable lattice L_7 with $4 + 9 + 2 + 3 = 216$ elements and with depth of $2 + 2 + 1 + 2 = 7$. If we consider buying three products again, we get $3 \cdot 7 + 1 = 22$ qualitative classes and we obtain four important classes:

Class **0**:

$$\langle 0, 0, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle 0, 0, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle 0, 0, \text{over 2 kg}, \text{Integrated} \rangle$$

Class **6**:

$$\langle \text{MS Windows}, \text{Lenovo}, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MS Windows}, \text{ASUS}, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MacOS}, \text{Apple}, \text{over 2 kg}, \text{Integrated} \rangle$$

Class **15**:

$$\langle \text{MS Windows}, \text{HP}, \text{do 2 kg}, \text{Game playing} \rangle$$
$$\langle \text{MS Windows}, \text{Dell}, \text{do 2 kg}, \text{Game playing} \rangle$$
$$\langle \text{MacOS}, \text{Apple}, \text{do 2 kg}, \text{Game playing} \rangle$$

Class **21**:

$$\langle 1, 1, \text{do 2 kg}, \text{Game playing} \rangle$$
$$\langle 1, 1, \text{do 2 kg}, \text{Game playing} \rangle$$
$$\langle 1, 1, \text{do 2 kg}, \text{Game playing} \rangle$$

An important fact is that the smallest available class is **6** and the largest available class is **15**. For every qualitative class lower than **6** or larger than **15**, it is true that any chosen product in this class contains unavailable parameter and therefore it is a sold out product. By this step, we have lost 12 qualitative classes. The new range of classes is between **6** – **15** to select the golden mean. Of course, even in this range we can find a product (more precisely, a combination of parameters) that cannot be purchased, such as ⟨MacOS, Acer, over 2 kg, Game playing⟩. We consider such products to be sold out and we try to find another suitable trio of available products in the given qualitative class (these products can be purchased).

For example, we can choose the golden mean as the class **11**. This qualitative class includes this three products:

⟨MS Windows, ASUS, up to 2 kg, Game playing⟩
⟨MS Windows, ASUS, up to 2 kg, Integrated⟩
⟨MS Windows, ASUS, up to 2 kg, Integrated⟩

or the following three products

⟨MacOS, Apple, over 2 kg, Game playing⟩
⟨MacOS, Apple, over 2 kg, Game playing⟩
⟨MacOS, Apple, over 2 kg, Dedicated⟩.

Now let us look at how to approach the situation if we convert distances to points, that is, a distance of length n will correspond to n points. We know that every incomparable parameter is worth 1 point, because the distance of such a parameter from the smallest element 0 in the corresponding lattice is equal to 1. In our case, we have two incomparable parameters in one product and we purchase three products. Therefore, it must be true that the three available products are initially worth 6 points, which exactly corresponds to the smallest available class. The other points of this trio of products depend only on comparable parameters. If we consider the class **11**, then we only have 5 points left, which must be divided between the parameters of the weight of the laptop and the type of the graphics card. Based on this idea and depending on the situation, we can decide what comparable parameters we prefer. More precisely, what is the most important for us in a given purchase.

Suppose that we want to have all three laptops lightweight (up to 2 kg), it will cost 3 points. The remaining 2 points must be allocated to the type of the graphics card. The potential result can be:

⟨MS Windows, ASUS, up to 2 kg, Dedicated⟩
⟨MS Windows, Acer, up to 2 kg, Dedicated⟩
⟨MacOS, Apple, up to 2 kg, Integrated⟩

or

⟨MS Windows, HP, up to 2 kg, Game playing⟩
⟨MS Windows, MSI, up to 2 kg, Integrated⟩
⟨MacOS, Apple, up to 2 kg, Integrated⟩,

where the second mentioned result (three products) contains a sold out product (the product cannot be bought in this variant of parameters) and therefore this result cannot be used. On the other hand, if we prefer the type of the graphics card, we can use all 5 points to this parameter. We have already seen a similar result, but to illustrate we present two more:

⟨MS Windows, Dell, over 2 kg, Game playing⟩
⟨MS Windows, Acer, over 2 kg, Game playing⟩
⟨MS Windows, Lenovo, over 2 kg, Dedicated⟩

or

⟨MS Windows, ASUS, over 2 kg, Game playing⟩
⟨MacOS, Apple, over 2 kg, Game playing⟩
⟨MS Windows, MSI, over 2 kg, Dedicated⟩,

where again the second mentioned result cannot be used because it contains a sold out product.

6. Finding the golden mean using a budget or priorities

At the end of Example 3.1, we looked at the budget and tried to determine the best possible qualitative class (golden mean). However, as we will show in the following example, including the budget in the calculation is not as easy as it may seem.

Example 6.1. Consider the same task as in Example 5.1. The goal is to determine the right qualitative class (golden mean) based on the preliminary budget. Suppose that the budget is set at 70 000 CZK.

Now it depends on the situation how the prices are allocated. Either for individual parameters or for the whole product (a combination of four products). We choose the case number two and determine the (lowest) prices of the purchased products for all possible combinations. Therefore, for example, the prices of individual products can be seen in the Table 5.

Immediately, we can see the problem. It is based on the fact that in real life it is not true that the best product (according to comparable parameters) is the most expensive one. That means, how we will show in a moment, different qualitative classes may have different prices and it is not always the true, that a higher qualitative class has a higher price (the total price of three products is higher).

With a budget of 70 000 CZK we can buy following products:

(1) Qualitative class: **15**. Price: 57 000 CZK.

$3 \times$ ⟨MS Windows, Acer, up to 2 kg, Game playing⟩

(2) Qualitative class: **14**. Price: 70 000 CZK.

⟨MS Windows, Lenovo, up to 2 kg, Game playing⟩
⟨MS Windows, ASUS, up to 2 kg, Game playing⟩
⟨MS Windows, HP, up to 2 kg, Dedicated⟩

(3) Qualitative class: **13**. Price: 70 000 CZK.

⟨MS Windows, Lenovo, up to 2 kg, Game playing⟩
⟨MS Windows, ASUS, up to 2 kg, Game playing⟩
⟨MS Windows, HP, up to 2 kg, Dedicated⟩

(4) Qualitative class: **12**. Price: 64 000 CZK.

⟨MS Windows, Dell, up to 2 kg, Game playing⟩
⟨MS Windows, Dell, up to 2 kg, Dedicated⟩
⟨MS Windows, Dell, up to 2 kg, Integrated⟩

(5) Qualitative class: **11**. Price: 64 000 CZK.

⟨MS Windows, HP, up to 2 kg, Dedicated⟩
⟨MS Windows, HP, over 2 kg, Dedicated⟩
⟨MS Windows, HP, over 2 kg, Game playing⟩

(6) Qualitative class: **10**. Price: 69 000 CZK.

⟨MS Windows, Lenovo, up to 2 kg, Dedicated⟩
⟨MS Windows, HP, over 2 kg, Dedicated⟩
⟨MacOS, Apple, up to 2 kg, Integrated⟩

(7) Qualitative class: **9**. Price: 63 000 CZK.

⟨MS Windows, MSI, over 2 kg, Game playing⟩
⟨MS Windows, Acer, over 2 kg, Integrated⟩
⟨MacOS, Apple, up to 2 kg, Integrated⟩

Table 5
The table showing the prices of laptops in all possible combinations in the Example 6.1.

Lenovo					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
7 000 CZK	15 000 CZK	33 000 CZK	16 000 CZK	41 500 CZK	18 000 CZK
ASUS					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
7 000 CZK	38 000 CZK	20 000 CZK	15 500 CZK	–	19 500 CZK
MSI					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
30 000 CZK	–	23 000 CZK	–	–	24 000 CZK
HP					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
12 500 CZK	17 000 CZK	43 000 CZK	30 000 CZK	26 000 CZK	21 000 CZK
Acer					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
7 500 CZK	14 500 CZK	19 000 CZK	11 000 CZK	50 000 CZK	19 000 CZK
Dell					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
13 500 CZK	20 000 CZK	30 500 CZK	–	70 000 CZK	24 000 CZK
Apple					
up to 2 kg			over 2 kg		
Integrated	Dedicated	Game playing	Integrated	Dedicated	Game playing
28 000 CZK	–	–	72 000 CZK	–	–

(8) Qualitative class: **8**. Price: 59 000 CZK.

$$\langle \text{MS Windows}, \text{ASUS}, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MS Windows}, \text{MSI}, \text{up to 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MS Windows}, \text{Dell}, \text{up to 2 kg}, \text{Integrated} \rangle$$

(9) Qualitative class: **7**. Price: 59 500 CZK.

$$\langle \text{MS Windows}, \text{Lenovo}, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MS Windows}, \text{HP}, \text{over 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MS Windows}, \text{Dell}, \text{up to 2 kg}, \text{Integrated} \rangle$$

(10) Qualitative class: **6**. Price: 33 000 CZK.

$$3 \times \langle \text{MS Windows}, \text{Acer}, \text{over 2 kg}, \text{Integrated} \rangle$$

We see that the chosen budget covers all available qualitative classes, so it is not possible to determine a qualitative class by budget. In Example 3.1 we took advantage of the requirements of the employees but now that will not help either as prices vary. Furthermore, it is not true that the price of products increases with the increasing qualitative class. As a result, the low class may not have low prices, such as products

$$3 \times \langle \text{MacOS}, \text{Apple}, \text{over 2 kg}, \text{Integrated} \rangle$$

belonging to the class **6** with a total value of 216 000 CZK. Therefore, we can say that according to this method, it is not possible to determine exactly the final qualitative class (golden mean). However, we can use it to get the highest possible qualitative class with a given budget. With an original budget of 70 000 CZK we covered all qualitative classes, so in this case, for example, we could find the cheapest purchase in the qualitative class **15**. The result would be the option (1) in the list above. If we set a new budget, for example, 35 000 CZK, we can try to find three products to meet this budget and belong to the highest possible qualitative class. By examining the tables of products with prices, we find that the highest possible qualitative class is **11**, because either products

$$\langle \text{MS Windows}, \text{ASUS}, \text{up to 2 kg}, \text{Game playing} \rangle$$
$$\langle \text{MS Windows}, \text{ASUS}, \text{up to 2 kg}, \text{Integrated} \rangle$$
$$\langle \text{MS Windows}, \text{Acer}, \text{up to 2 kg}, \text{Integrated} \rangle$$

cost 34 500 CZK or products

$$\langle \text{MS Windows}, \text{Acer}, \text{up to 2 kg}, \text{Game playing} \rangle$$
$$2 \times \langle \text{MS Windows}, \text{ASUS}, \text{up to 2 kg}, \text{Integrated} \rangle$$

cost 33 000 CZK belong to the class **11**. Therefore, the final purchase should belong to the class **11** and its total value should not exceed the price of 35 000 CZK.

At the end of the example, we will show one more method to determine the resulting qualitative class—by our priorities for purchased products. Let us imagine we want to buy three laptops and we have some idea of what the parameters of laptops should be. Based on our idea, we can put together a combination of three products and find out what class these products belong to. It should be noted here that we must not cling too much to our preferences, because in that case, we would only like the purchase we want and no other purchase would be considered. The aim of this method is to determine the right qualitative class (golden mean) on the basis of some requirement. After that, we can find potentially better variant in this class than our idea. To illustrate, our priorities could be in the form of:

$$\langle \text{MS Windows}, \text{Dell}, \text{over 2 kg}, \text{Game playing} \rangle$$
$$\langle \text{MS Windows}, \text{MSI}, \text{up to 2 kg}, \text{Dedicated} \rangle$$
$$\langle \text{MacOS}, \text{Apple}, \text{up to 2 kg}, \text{Integrated} \rangle.$$

Note that our priorities do not have to be examined with regard to sold-out products, as this is only a potential purchase, not a final one. Among the products representing our priorities, the second product is sold out because it cannot be purchased. For this reason, this combination of products cannot be final purchase. However, we can determine the appropriate qualitative class. We simply find that these three products belong to the qualitative class **11**. At this point, we already have everything we need to make a list of all the possible available combinations of the three products belonging to the qualitative class **11** and choose the combination that most closely matches our priorities. For the final choose, we will use a situation from a real life, where the boss has a meeting with three of his employees regarding the purchase of three laptops. The description of the meeting is as follows:

Employee number 3 is stubborn and wants to have an Apple laptop at all costs. Since the tables of prices of products have no other possible option, we must keep the product for this employee. Employee number 3 is leaving the meeting. Employee number 1 is new and shy, so employee number 2 does not hesitate and tells us that insists on the brand MSI of the laptop and also needs a powerful graphics card to work. Again, according to the table, we see that the only option is to move to a better graphics card, so employee 2 gets a laptop with a game playing graphics card. Employee number 2 leaves the meeting. Whereas employee number 2 got something better, employee number 1 must get something worse. He does not care about the weight of the laptop, but he would like a good graphics card. He can no longer get a game playing graphics card, but dedicated graphics card can be. However, a new problem has emerged. The cost of such a laptop would be 70 000 CZK. Eventually, he changes his mind about the laptop brand and moves from Dell to Lenovo. The meeting is over.

The final purchase for our three employees is as follows:

$$\langle \text{MS Windows}, \text{Lenovo}, \text{over 2 kg}, \text{Dedicated} \rangle$$
$$\langle \text{MS Windows}, \text{MSI}, \text{up to 2 kg}, \text{Game playing} \rangle$$
$$\langle \text{MacOS}, \text{Apple}, \text{up to 2 kg}, \text{Integrated} \rangle$$

for a total value of 92 500 CZK.

7. Use of other types of lattices

Until now, we only had two types of lattices: a chain and a n-dimensional diamond. However, we can get any type of lattice depending on the situation. We will find out later that transfer-stable lattices do not give as good information as distance-stable lattices. Therefore, it will be advantageous to work only with distance-stable lattices. Before approaching the problem with transfer-stable lattices, we will show a theoretical example of when lattices of other types may appear.

Example 7.1. The aim of this example is to buy four devices of the same type, where one device is composed of two separable parts. Each part has advantages and disadvantages depending on the use or location of the device. The first part has the advantages A, B and the disadvantages C, D and the second part has the advantages a, b, c and the disadvantages d, e, f. Both parts can be produced in the combinations indicated in the Table 6.

Both parts (combinations of advantages and disadvantages) can be represented in the following (distance-stable) lattices (see Fig. 5) according to the Table 6.

By the direct product of these lattices we obtain the lattice L_8 with $4 \cdot 5 = 20$ elements and depth of $2 + 3 = 5$. For the purchase of four devices we get $4 \cdot 5 + 1 = 21$ qualitative classes.

In this case, we cannot get sold out products because it is possible to produce any combination (as in the introductory example). In other words, all parts listed in the Table 6 can be made and also any device composed of these parts can be build. That means, the smallest available class is the smallest class **0**, which is determined by the quadruple $\langle v_1u_1, v_1u_1, v_1u_1, v_1u_1 \rangle$ and the highest available class is the highest class **20**, which is determined by the quadruple $\langle v_4u_5, v_4u_5, v_4u_5, v_4u_5 \rangle$. To illustrate, the quadruple $\langle v_1u_3, v_4u_1, v_3u_2, v_2u_4 \rangle$ belonging to the qualitative class **8**[13] represents four devices, where the first device consists of two parts v_1 and u_3 such that the first part has disadvantages C, D and the other part has the advantage b and disadvantages e, f. Further, the fourth device is composed of two parts v_2 and u_4, where the first part has the advantage A and the disadvantage C and the second part has the advantages a, b and the disadvantage f.

Table 6
Advantages and disadvantages of both parts of the device for Example 7.1 and possible production combinations of the whole device.

First part	Advantages					Disadvantages				
	A	B	a	b	c	C	D	d	e	f
v_4	×	×								
v_3		×					×			
v_2	×					×				
v_1						×	×			
Second part	A	B	a	b	c	C	D	d	e	f
u_5			×	×	×					
u_4			×	×						×
u_3				×					×	×
u_2			×					×	×	
u_1								×	×	×

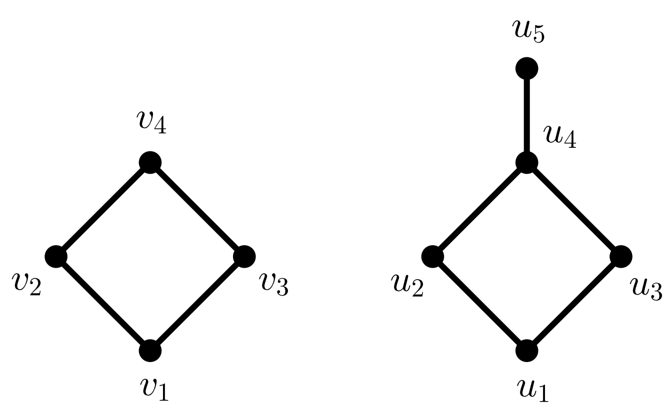

Fig. 5. Distance-stable lattices representing the advantages and disadvantages of the first and second part of the device, respectively.

Table 7
The requirements for the advantages of individual parts for four devices.

	Advantages				
	A	B	a	b	c
First device	•		•		
Second device		•		•	
Third device			•	•	
Fourth device					

As in the previous example, we can set priorities for the purchase of four devices. Depending on the situation, the minimum requirements for the advantages of individual parts of all devices of the quadruple may be as follows (see Table 7).

By a simple analysis of this situation, we find that the minimum and the maximum qualitative class that meets our requirements is **6** and **20**, respectively. Obviously, there is only one quadruple $\langle v_2u_2, v_3u_3, v_1u_4, v_1u_1 \rangle$ meeting our requirements in the minimum class **6**. Similarly, there is only one quadruple $\langle v_4u_5, v_4u_5, v_4u_5, v_4u_5 \rangle$ meeting our requirements in the maximum class **20**. The last fact is not based

[13] (1) $d(v_1, v_1) = 0, d(u_1, u_3) = 1 \to 0 + 1 = 1$ (2) $d(v_1, v_4) = 2, d(u_1, u_1) = 0 \to 2 + 0 = 2$ (3) $d(v_1, v_3) = 1, d(u_1, u_2) = 1 \to 1 + 1 = 2$ (4) $d(v_1, v_2) = 1, d(u_1, u_4) = 2 \to 1 + 2 = 3 \Rightarrow 1 + 2 + 2 + 3 = 8$.

on our requirements (as was the case with the class **6**) but on the fact that the highest (even the smallest) qualitative class always has only one element. If we look at the class **7**, we have 11 options that meet our requirements, i.e.,

1. $\langle v_4u_2, v_3u_3, v_1u_4, v_1u_1 \rangle$
2. $\langle v_2u_4, v_3u_3, v_1u_4, v_1u_1 \rangle$
3. $\langle v_2u_2, v_4u_3, v_1u_4, v_1u_1 \rangle$
4. $\langle v_2u_2, v_3u_4, v_1u_4, v_1u_1 \rangle$
5. $\langle v_2u_2, v_3u_3, v_2u_4, v_1u_1 \rangle$
6. $\langle v_2u_2, v_3u_3, v_3u_4, v_1u_1 \rangle$
7. $\langle v_2u_2, v_3u_3, v_1u_5, v_1u_1 \rangle$
8. $\langle v_2u_2, v_3u_3, v_1u_4, v_2u_1 \rangle$
9. $\langle v_2u_2, v_3u_3, v_1u_4, v_3u_1 \rangle$
10. $\langle v_2u_2, v_3u_3, v_1u_4, v_1u_2 \rangle$
11. $\langle v_2u_2, v_3u_3, v_1u_4, v_1u_3 \rangle$.

It depends on the circumstances to which qualitative class we can advance. Assume a scenario we have a budget of 8 000 000 CZK for purchase four devices. Individual qualitative classes (regardless of the specific device) have prices set as follows (the price is for the purchase of four devices in the given class):

0. 10 000 CZK
1. 20 000 CZK
2. 40 000 CZK
3. 80 000 CZK
4. 160 000 CZK
5. 320 000 CZK
6. 640 000 CZK
7. 1 280 000 CZK
8. 2 560 000 CZK
9. 5 120 000 CZK
10. 10 240 000 CZK
11. 20 480 000 CZK
⋮
18. 2 621 440 000 CZK
19. 5 242 880 000 CZK
20. 10 485 760 000 CZK.

We found that the minimum qualitative class meeting our requirements is the class **6** with a total cost of 640 000 CZK for all four devices. We can advance up to the class **9** with a total price of 5 120 000 CZK due to the budget. Of course, we cannot use the following class **10** because its price exceeds the specified budget. Based on this idea, we can choose four devices from the class **9**. We have the following option:

$\langle v_2u_2, v_3u_3, v_4u_4, v_2u_1 \rangle$
$\langle v_2u_2, v_3u_3, v_2u_4, v_3u_3 \rangle$
$\langle v_2u_2, v_3u_3, v_3u_4, v_4u_1 \rangle$
$\langle v_2u_2, v_3u_3, v_1u_4, v_2u_4 \rangle$

or we can improve our requirements and we get:

$\langle v_4u_4, v_3u_3, v_1u_5, v_1u_1 \rangle$
$\langle v_4u_2, v_4u_3, v_1u_5, v_1u_1 \rangle$
$\langle v_2u_5, v_3u_4, v_1u_4, v_1u_1 \rangle$
$\langle v_2u_2, v_4u_5, v_1u_4, v_1u_1 \rangle$.

In this example, we can see that other types of lattices than just chains and n-dimensional diamonds can be used effectively. However, we must be careful what types of lattices we use, because in the paper [26] it was shown that this theory of transfer-stable functions can only be applied to transfer-stable lattices. Even these lattices are not effective to solve similar tasks as before. That means, if we obtain a transfer-unstable lattice (for example, see Fig. 6) in the task of the example, then the theory of transfer-stable aggregation functions cannot be applied to this lattice because the qualitative classes would not be linearly ordered. Therefore we could not say which class is better or worse.

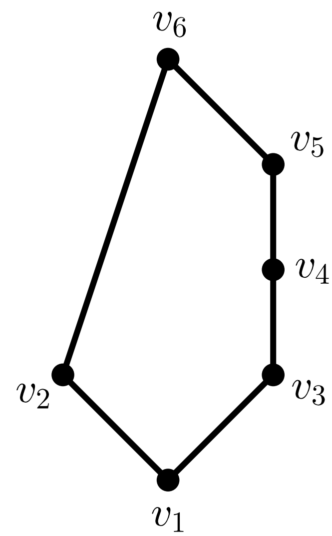

Fig. 6. The example of a prohibited lattice for the application of transfer-stable aggregation functions.

8. Transfer-stable lattices as an inappropriate assignment

At the end of the previous section, we mentioned that the use of transfer-unstable lattices is not allowed. However, even transfer-stable distance-unstable lattices do not give good information during the calculation. In the paper [26] it was shown that the central block $[(0,1)]$ (respectively $[(0,\dots,0,1)]$)[14] contains most pairs (n-tuples) of a direct product. As a result, most options (purchases) belong to the same class. This case is very similar to the Example 5.1, where all available options belonged to a one class. The following example shows the use of transfer-stable distance-unstable lattices.

Example 8.1. In this example, our goal is to buy two packages containing a mobile phone, a laptop, a smart watch, and a tablet. We have the products listed in the Table 8.

Relevant lattices (horizontal sums): A – the lattice of mobile phones, B – the lattice of laptops, C – the lattice of smart watches, D – the lattice of tablets, for products from the Table 8 are depicted in Fig. 7.

We have mentioned several times before that a corrected assignment of the example is the key to success and quality results. Unfortunately, we will not avoid problems this time either.

The first problem is that the lattice of mobile phones is not a transfer-stable lattice. According to the note at the beginning of this section, this lattice (assignment) is not possible. One possible solution to this problem is to remove (or add) an element from the lattice A. For example, by removing the element "Galaxy A" we get a transfer-stable lattice A' (see Fig. 8).

The second problem is the quality of the assignment with regard to the placement of products in the lattice. Specifically, there is a problem with the number of points (distance from the smallest element). For example, we can see that either the "Acer Predator" costs more points than the "MacBook Pro" or the "Apple Watch Series 7" has more points than the "Garmin Fenix". Of course, the compilation of the assignment depends on our preferences and the situation. Nevertheless, it would be

[14] The smallest element in the lattice is 0 and the largest element in the lattice is 1.

Table 8
List of products for the package in the Example 8.1.

MOBILE PHONES			
Apple	Samsung	Xiaomi	Realme
iPhone 13	Galaxy Z	Mi	8
iPhone 12	Galaxy S	Redmi	7
iPhone 11	Galaxy M	–	–
iPhone SE	Galaxy A	–	–
LAPTOPS			
Apple	HP	Lenovo	Acer
MacBook Pro	EliteBook	Legion	Predator
MacBook Air	Spectre	ThinkPad	Nitro
–	ProBook	ThinkBook	Spin
–	–	Yoga	Extensa
–	–	–	Aspire
SMART WATCHES			
Apple	Samsung	Xiaomi	Garmin
Series 7	Galaxy Watch	Mi Watch	Fenix
Series 6	–	Mi Band	Vivoactive
SE	–	–	Venu
Series 3	–	–	–
TABLETS			
Apple	Samsung	Lenovo	Huawei
iPad Pro	Tab S	Yoga Tablet	MatePad
iPad Air	Tab A	Tab	MediaPad

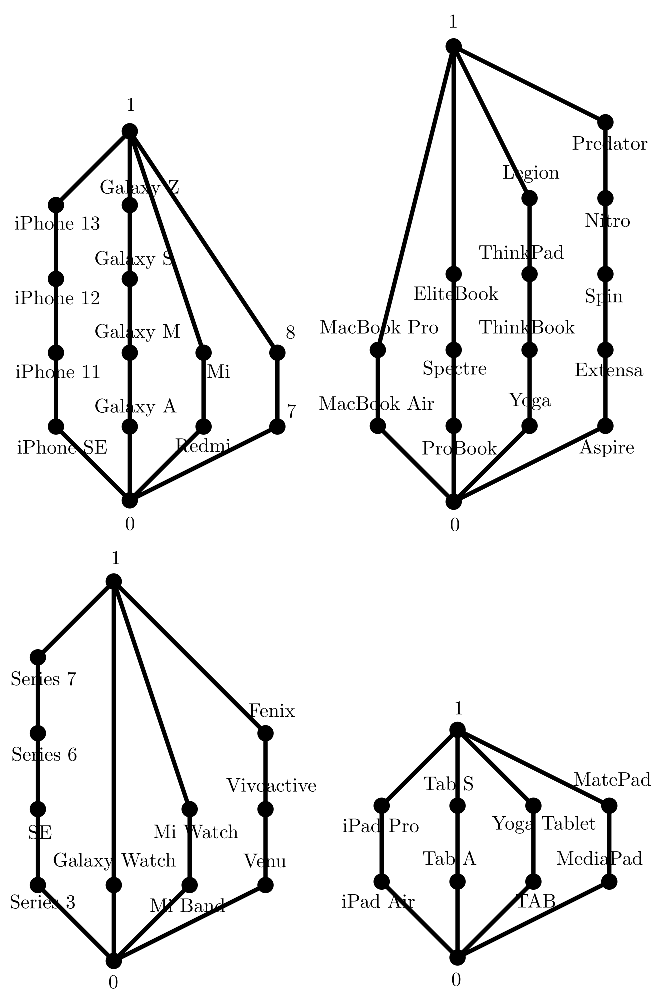

Fig. 7. The horizontal sums A, B, C and D representing the types of mobile phones, laptops, smart watches and tablets, respectively.

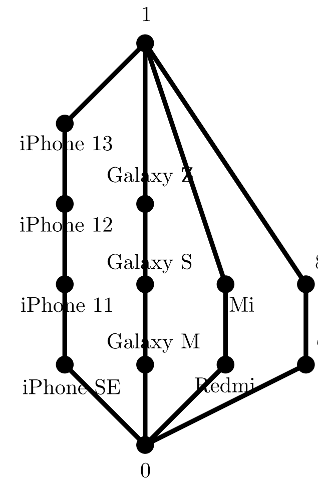

Fig. 8. Transfer-stable lattice A' of mobile phones.

wise for specific products to have the same number of points. We will eliminate this problem later, because at the moment the third problem is much more serious. It depends on the very essence of transfer-stable lattices.

We obtain the lattice L_9 with $13 \cdot 16 \cdot 12 \cdot 10 = 24\,960$ elements by the direct product of the lattices A', B, C and D. However, in the lattice L_9, only 12 320 quadruplets are valid, i.e., they do not contain unavailable parameters 0 and 1.

Now we need to solve the number of classes to purchase two packages. For transfer-stable lattices, the first unstable elements relative to 0 and 1 are important elements. Since the lattices A', B, C and D are horizontal sums, then the first unstable elements are the elements 0 and 1. It can be proved that the direct product of transfer-stable lattices does not change the number of classes, i.e., the number of classes of direct product will be the same as the smallest number of classes of lattices A', B, C (lattice D is not used because it is distance-stable lattice). The result is the lattice C whose the number of classes is 5, because the individual classes of this lattice are $[\langle 0,0\rangle]$, $[\langle \text{Series } 3, 0\rangle]$, the central block $[\langle 0,1\rangle]$, $[\langle 1, \text{Series } 7\rangle]$ and $[\langle 1,1\rangle]$. The lattice L_9 has the same number of classes, i.e.,

0 … $[\langle 0,0,0,0\rangle\,;\langle 0,0,0,0\rangle]$
1 … $[\langle 0,0,\text{Series } 3,0\rangle\,;\langle 0,0,0,0\rangle]$
2 … $[\langle 0,0,1,0\rangle\,;\langle 0,0,0,0\rangle]$
3 … $[\langle 1,1,\text{Series } 7,1\rangle\,;\langle 1,1,1,1\rangle]$
4 … $[\langle 1,1,1,1\rangle\,;\langle 1,1,1,1\rangle]$.

Qualitative classes **0**, **1**, **3** and **4** are unusable because any 4-tuple always contains an unavailable parameter 0 or 1. All available 4-tuples, such as

$$\langle \text{Galaxy Z}, \text{Yoga}, \text{Vivoactive}, \text{Tab A}\rangle$$
$$\langle \text{iPhone 11}, \text{MacBook Air}, \text{Series 6}, \text{iPad Air}\rangle$$

belong to the central block **2**. Thus, there is no distribution of purchases of products in the various qualitative classes, as we have seen in the previous examples. In conclusion, we receive that even transfer-stable lattices are not a suitable option for calculating the purchase of several products.

9. Transformation of a transfer-stable lattice into a distance-stable lattice

Now, we know that the problem is not only transfer-unstable lattices, but also transfer-stable ones. Hence, we show how to prevent

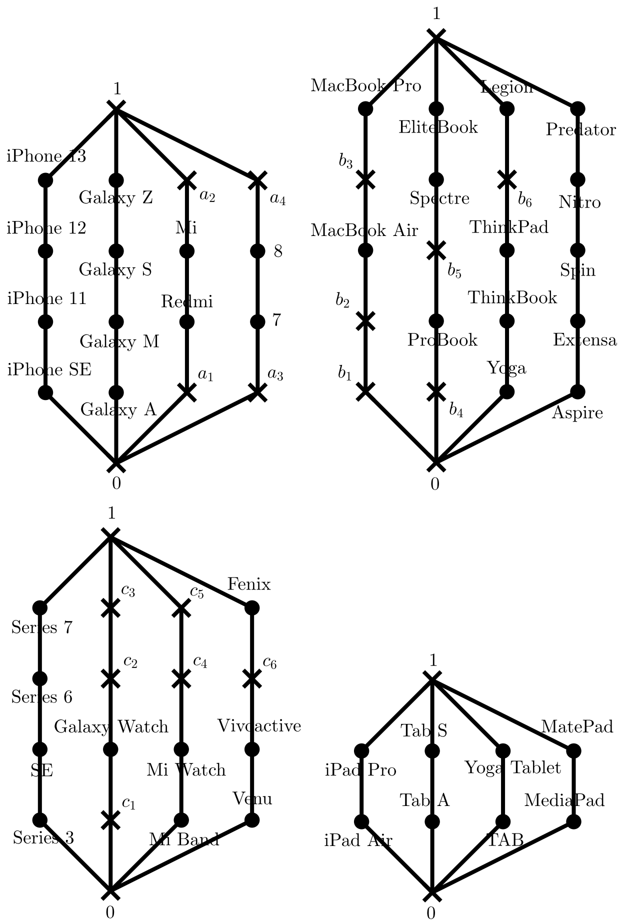

Fig. 9. The distance-stable lattices A_2, B_2, C_2 and D_2 representing the types of mobile phones, laptops, smart watches and tablets, respectively.

this problem. One way to solve this problem is to add elements to the lattice to create a distance-stable lattice. In this way, we could also avoid problem number two, mentioned in the previous example, and thus create an assignment that would better reflect reality. Therefore, we will try to solve once again the example from the previous section by adding elements to transfer-stable lattices to obtain distance-stable lattices.

Example 9.1. Consider the same assignment as in Example 8.1. This time, we will add new auxiliary elements to the lattices (horizontal sums) A, B, C and D. We get the distance-stable lattices (horizontal sums) A_2, B_2, C_2 and D_2 depicted in Fig. 9. The new elements $a_1, a_2, \dots, b_1, b_2, \dots, c_5, c_6$ represent unavailable parameters (as the parameters 0 and 1). Thanks to these elements (parameters), we have balanced the original products in the individual lattices into similar price (qualitative) categories.

By the direct product of the distance-stable lattices A_2, B_2, C_2 and D_2 we obtain the lattice L_{10} with $18 \cdot 22 \cdot 18 \cdot 10 = 71\,280$ elements. However, only 12 320 elements are usable because they do not contain unavailable parameters. The depth of the lattice L_{10} is $5+6+5+3 = 19$. Thus, if we want to buy two packages, the final number of qualitative classes is $2 \cdot 19 + 1 = 39$, where

0 … The smallest class

$\langle 0,0,0,0 \rangle$
$\langle 0,0,0,0 \rangle$,

8 … The smallest available class

$\langle$iPhone SE, Yoga, Series 3, iPad Air$\rangle$
$\langle$Galaxy A, Aspire, Venu, Tab A$\rangle$,

30 … The highest available class

$\langle$iPhone 13, MacBook Pro, Series 7, iPad Pro$\rangle$
$\langle$Galaxy Z, EliteBook, Fenix, Tab S$\rangle$,

38 … The highest class

$\langle 1,1,1,1 \rangle$
$\langle 1,1,1,1 \rangle$.

Of course, combinations with unavailable parameters may appear. To illustrate, combination from class **19**

$\langle$Mi, Extensa, Galaxy Watch, Yoga Tablet$\rangle$
$\langle$7, Spectre, Series 6, Media Pad$\rangle$,

can be purchased, but a combination from the same class

$\langle a_1$, MacBook Air, $c_1, 1\rangle$
$\langle a_4, b_5$, Vivoactive, iPad Pro$\rangle$,

or

$\langle a_3, b_4, c_5, 0 \rangle$
$\langle a_2, b_5, c_2, 1 \rangle$,

cannot be used because at least one parameter is unavailable.

Now we could choose the qualitative class as the golden mean and find the combination that is best for us. This time we will do it differently. As in the previous examples, we will try to set the prices of individual products and select the appropriate qualitative class according to the budget or priorities. The chosen prices are listed in Table 9.

According to the prices in the tables, we can find the cheapest and most expensive possible combination in the smallest and highest available class.

Class **8**:

- The cheapest combination with a total value of 32 000 CZK is:

  $2\times\langle$Galaxy A, Aspire, Mi Band, Tab$\rangle$

- The most expensive combination with a total value of 102 000 CZK is:

  $2\times\langle$iPhone SE, Yoga, Series 3, iPad Air$\rangle$

Class **30**:

- The cheapest combination with a total value of 126 000 CZK is:

  $2\times\langle$iPhone 13, Legion, Series 7, Yoga Tablet$\rangle$

- The most expensive combination with a total value of 224 000 CZK is:

  $2\times\langle$Galaxy Z, Predator, Fenix, iPad Pro$\rangle$

Based on this information, we can say that if the budget is greater than 126 000 CZK, we will choose purchase from any qualitative class. The higher class, the better, but it is not necessary.

Let us choose a budget of 70 000 CZK. Then, for example, the possible combinations are:

Table 9
The prices of mobile phones, laptops, smart watches and tablets for Example 9.1.

MOBILE PHONES			
Apple		Samsung	
iPhone 13	20 000 CZK	Galaxy Z	27 000 CZK
iPhone 12	17 000 CZK	Galaxy S	14 000 CZK
iPhone 11	14 500 CZK	Galaxy M	4 500 CZK
iPhone SE	12 000 CZK	Galaxy A	3 000 CZK
Xiaomi		Realme	
Mi	8 000 CZK	8	4 500 CZK
Redmi	6 000 CZK	7	4 500 CZK
LAPTOPS			
Apple		HP	
MacBook Pro	34 000 CZK	EliteBook	28 500 CZK
MacBook Air	28 000 CZK	Spectre	26 000 CZK
–	–	ProBook	17 000 CZK
Lenovo		Acer	
Legion	23 000 CZK	Predator	50 000 CZK
ThinkPad	18 000 CZK	Nitro	20 000 CZK
ThinkBook	16 000 CZK	Spin	12 000 CZK
Yoga	16 000 CZK	Extensa	10 500 CZK
–	–	Aspire	9 000 CZK
SMART WATCHES			
Apple		Samsung	
Series 7	11 000 CZK	Galaxy Watch	7 000 CZK
Series 6	9 500 CZK	–	–
SE	8 000 CZK	–	–
Series 3	6 000 CZK	–	–
Xiaomi		Garmin	
Mi Watch	3 000 CZK	Fenix	12 000 CZK
Mi Band	1 000 CZK	Vivoactive	6 500 CZK
–	–	Venu	4 500 CZK
TABLETS			
Apple		Samsung	
iPad Pro	23 000 CZK	Tab S	10 000 CZK
iPad Air	17 000 CZK	Tab A	4 500 CZK
Lenovo		Huawei	
Yoga Tablet	9 000 CZK	MatePad	11 500 CZK
Tab	3 000 CZK	MediaPad	4 000 CZK

- Qualitative class: **8**. Price: 58 000 CZK.

  $\langle$iPhone SE, Aspire, Venu, Tab A$\rangle$
  $\langle$Galaxy A, Yoga, Series 3, TAB$\rangle$

- Qualitative class: **11**. Price: 69 500 CZK.

  $\langle$iPhone 11, ThinkBook, Vivoactive, Tab A$\rangle$
  $\langle$Galaxy A, Yoga, Series 3, TAB$\rangle$

- Qualitative class: **14**. Price: 66 500 CZK.

  $\langle$iPhone 11, ThinkBook, Vivoactive, Tab A$\rangle$
  $\langle$Galaxy M, Extensa, Galaxy Watch, TAB$\rangle$

- Qualitative class: **16**. Price: 67 000 CZK.

  $\langle$7, Extensa, Mi Watch, Yoga Tablet$\rangle$
  $\langle$Redmi, ThinkBook, SE, Tab S$\rangle$

- Qualitative class: **22**. Price: 65 000 CZK.

  $\langle$8, Spin, Mi Watch, Yoga Tablet$\rangle$
  $\langle$8, Spin, Series 7, Yoga Tablet$\rangle$

- Qualitative class: **22**. Price: 69 000 CZK.

  $\langle$8, ThinkPad, Fenix, TAB$\rangle$
  $\langle$8, Spin, Fenix, TAB$\rangle$

From the list above, we see that the maximum possible qualitative class is **22** for the selected budget. At the moment, it is up to us whether we stay in this class or choose a lower class, where there are more options (combinations) to make a purchase with the specified budget. The result could be one of above mentioned combinations for class **22**, if we wanted to stay in this class.

Another possible priority (depending on the budget) could be the choice of a certain type of device. For example, consider that the first package should contain the best laptop and the second package should include an Apple mobile phone. Then the resulting combination could be:

$\langle$8, Legion, Galaxy Watch, Media Pad$\rangle$
$\langle$iPhone 11, Aspire, Venu, Tab$\rangle$,

and belongs to the class **16** with a total value of 69 500 CZK. We see that each new priority reduces the number of options in the class and, more importantly, we get closer to the resulting qualitative class with the next requirement. As a result, it is up to us which direction we take. Either we prefer price and quality, so we want to get to the highest possible quality class in that case, or our main priority will not be price, and we try to get to the qualitative class that best meets our requirements.

However, it may also happen that the next request will be too strong and we will not get any final solution. In particular, if we demand both packages composed of Apple products, then we will not get any suitable combination for a budget of 70 000 CZK. The cheapest Apple combination

$2 \times \langle$iPhone SE, MacBook Air, Series 3, iPad Air$\rangle$

belongs to the class **12** with a total price of 126 000 CZK, which is almost double the value of our budget.

In addition, this is "only" the class **12** compared to our highest found class **22**. For instance, the Apple set belonging to the class **22** is:

$\langle$iPhone 13, MacBook Pro, Series 7, iPad Pro$\rangle$
$\langle$iPhone 11, MacBook Air, Series 3, iPad Air$\rangle$

with a total price of 153 500 CZK.

10. Conclusion

The aim of the paper was to provide some solutions for the purchase of either several products or one product depending on several parameters. To do this, we used transfer-stable aggregation functions on finite lattices. As it turned out, the transfer-stable aggregation functions divided our potential purchases into several classes depending on the price:quality ratio. On the other hand, by using finite lattices (in particular, horizontal sums), we can compare the quality of products very elegantly and construct the most realistic assignment. Combining these two steps, we have a very good business strategy.

In an introductory example, we presented the theory of transfer-stable aggregation functions and showed the various applications it currently offers. Using this particular (very simple) example, we have outlined what we could expect in the paper. In the following example, we answered the question: "What happens when there is no product (combination) in the direct product?" The answer was that we considered these combinations and calculated with them as we normally would, but they were not allowed to appear in any form of result. We got the so-called *sold-out product*. It was a product that could not be purchased in a given combination of parameters (it did not exist), or it contained an unrealistic parameter (it was an element of lattice that did not represent anything real – *unavailable parameter*). The next part was significant important as we showed that it is not always appropriate to use chains. The reason was that we could not declare which parameter was better or worse. Therefore, we ordered them into the so-called *n*-dimensional diamond. Subsequently, it turned out that this was the right step, as the results better reflected the real situation. In the sixth section, we focused more on the actual search for the golden mean (the appropriate quality class). We have used not only the budget but also the priorities we have set for this examination. Therefore, the purpose of this section was not to find the final combination of products (parameters) but to provide a hint as to look for the class (golden mean) for this final combination. From the following chapter onwards, we have focused on different types of lattices. The intention was to show that any finite lattice can be used in the real world. Section 7 was very different from all the others. Here we have shown another possible application usability of these functions on various finite lattices. The last two sections were very similar. We set realistic conditions and tried to find the best possible solution. However, we found a problem in Section 8. Transfer-stable lattices are not a good choice for parameter ordering. We have found the problems that these lattices have. The main problem was that transfer-stable lattices have very few blocks (classes), which leads to uninteresting results. Only the final ninth section outlined the true potential of these functions. We have shown that ordering the parameters (products) into a distance-stable lattice (more precisely, into a horizontal sum) forms a very efficient assignment. This assignment can avoid unintended problems, such as product inconsistency in terms of quality.

As we mentioned in the introduction, we obtained most of the data from the website alza.cz. This also set the direction of the paper. When a product could not be bought in this store, we marked it as a sold-out product. If this shop recommended that the two products we selected were almost the same (in terms of price:quality ratio), we had to adjust the assignment accordingly. Thus, we used horizontal sums.

The main contribution of this paper was to help with the difficult decision to purchase several products using a very simple property such as transfer-stability. The first important result was that finite lattices (so far, mainly distance-stable lattices) could be used very efficiently for real situations. We have shown how to use these lattices to correctly build the assignment to match the real scenario as closely as possible. The second important fact was that the transfer-stable aggregation functions ignored uninteresting elements (artificially added elements to get a distance-stable lattice) in the lattices so that it did not affect the result in any way. Despite the strong relation between transfer-stable aggregation functions and transfer-stable lattices shown in previous papers, the last and most important fact was that these lattices do not reflect reality as well as the distance-stable lattices. Although this result was negative, it was also very important because these lattices have a very good property in mathematics—their blocks are linearly ordered. However, these functions did not have good features in the economy. We have shown that these functions are not useful.

Research for this paper can continue in several possible ways. First, other potential applications of transfer-stable aggregation functions can be shown (not only in economics). Second, the actual specification of real situations can be studied. More precisely, how best to set the task so that the results, as accurately as possible, match real-world experience. It is also possible to examine the golden mean and invent a procedure that would involve the use of transfer-stable functions. The following future direction of this paper is related to the previous possible solution. It is about programming the whole process (from input to output) so that it can be better used in practice. Finally, as far as the theoretical basis is concerned, it would be possible to further develop either the properties of distance-stable lattices in the theory of lattices or to show the advantages of transfer-stable distance-unstable lattices. It is also possible to study the transfer-stable functions themselves without the aggregation functions.

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

No data was used for the research described in the article.

References

[1] G. Beliakov, H. Bustince, T. Calvo, A Practical Guide to Averaging Functions, Springer, 2016.
[2] P. Pasteczka, On a lattice-like property of quasi-arithmetic means, J. Math. Anal. Appl. 486 (1) (2020).
[3] L. Rozovsky, Comparison of arithmetic, geometric, and harmonic means, Math. Not. 110 (1–2) (2021) 118–125.
[4] M.Y. Xie, J. Zhang, Inequalities for arithmetic–geometric–harmonic means of sector matrices, Linear Multilinear Algebra 70 (20) (2022) 5922–5931.
[5] AFRL de Hierro, C. Roldan, H. Bustince, J. Fernandez, I. Rodriguez, H. Fardoun, J. Lafuente, Affine construction methodology of aggregation functions, Fuzzy Sets Syst. 414 (2021) 146–164.
[6] B. De Baets, H. De Meyer, Maximal directions of monotonicity of an aggregation function, Fuzzy Sets and Systems 433 (2022) 54–78.
[7] S. Doria, R. Mesiar, A. Seliga, Sub-additive aggregation functions and their applications in construction of coherent upper previsions, Mathematics 9 (1) (2021).
[8] L.S. Jin, R. Mesiar, M. Kalina, J. Spirkova, S. Borkotokey, Generalized phi-transformations of aggregation functions, Fuzzy Sets and Systems 372 (2019) 124–141.
[9] R. Mesiar, A. Kolesarova, S. Borkotokey, L. Jin, Möbius product-based constructions of aggregation functions, Fuzzy Sets and Systems 448 (2022) 17–34, Special Issue.
[10] A. Seliga, K. Hrinakova, I. Seligova, Positively homogeneous and super-/sub-additive aggregation functions, Fuzzy Sets and Systems 451 (2022) 385–397.
[11] Y. Su, W. Zong, R. Mesiar, Characterization of homogeneous and quasi-homogeneous binary aggregation functions, Fuzzy Sets and Systems 433 (2022) 96–107.
[12] C. Bejines, M.J. Chasco, J. Elorza, Aggregation of fuzzy subgroups, Fuzzy Sets and Systems 418 (2021) 170–184, Special Issue.
[13] M. Couceiro, D. Dubois, H. Fargier, M. Grabisch, H. Prade, A. Rico, New directions in ordinal evaluation: Sugeno integrals and beyond, in: M. Doumpos, J. Figueira, Greco S., C. Zopounidis (Eds.), New Perspectives in Multiple Criteria Decision Making. Multiple Criteria Decision Making, Springer, Cham, 2019, 978-3-030-11481-7.
[14] R. Halaš, Z. Kurač, R. Mesiar, J. Pócs, Binary generating set of the clone of idempotent aggregation functions on bounded lattices, Inform. Sci. 462 (2018) 367–373.
[15] R. Liang, X. Zhang, Pseudo general overlap functions and weak inflationary pseudo BL-algebras, Mathematics 10 (16) (2022).
[16] G. Grätzer, F. Wehrung, Lattice Theory: Special Topics and Applications, Vol. 1, Springer, Switzerland, 2014.
[17] R. Halaš, J. Pócs, J. Pócsová, On join-dense subsets of certain families of aggregation functions, Mathematics 11 (1) (2023).
[18] I. Chajda, R. Halas, R. Mesiar, On the decomposability of aggregation functions on direct products of posets, Fuzzy Sets and Systems 386 (2020) 25–35.
[19] A. Mehenni, L. Zedam, B. Benseba, f-aggregation operators on a bounded lattice, Azerb. J. Math. 12 (1) (2022) 109–128.
[20] R. Mesiar, A. Kolesarova, T. Senapati, Aggregation on lattices isomorphic to the lattice of closed subintervals of the real unit interval, Fuzzy Sets and Systems 441 (2022) 262–278.
[21] R.E.B. Paiva, B.R.C. Bedregal, General pseudo quasi-overlap functions on lattices, Axioms 11 (8) (2022).
[22] Y. Sun, B. Pang, S.Y. Zhang, Binary relations induced from quasi-overlap functions and quasi-grouping functions on a bounded lattice, Comput. Appl. Math. 41 (8) (2022).
[23] Y. Wang, B.Q. Hu, On ordinal sums of overlap and grouping functions on complete lattices, Fuzzy Sets and Systems 439 (2022) 1–28, Special Issue.
[24] Y.Q. Zhang, H.W. Liu, New constructions of lattice-valued quasi-overlap functions, Iran. J. Fuzzy Syst. 20 (1) (2023).
[25] Z. Kurač, Transfer-stable means on finite chains, Fuzzy Sets and Systems 372 (2019) 111–123.
[26] Z. Kurač, T. Riemel, L. Rýparová, Transfer-stable aggregation functions on finite lattices, Inform. Sci. 521 (2020) 88–106.
[27] J. Dou, B. Ye, Product differentiation, exclusivity, and multi-purchasing, Ann. Econ. Finance 19 (1) (2018) 301–318.
[28] Z. Liu, T. Nishi, Analyzing just-in-time purchasing strategy in supply chains using an evolutionary game approach, J. Adv. Mech. Des., Syst., Manuf. 14 (5) (2020).
[29] G. Ma, J. Zheng, J. Wei, S. Wang, Y. Han, Robust optimization strategies for seller based on uncertainty sets in context of sequential auction, Appl. Math. Comput. 390 (2021).
[30] W.-W. Ren, Y. Liu, Q. Zhang, X.-Y. Wang, Sales strategies considering consumer purchasing preferences for imperfect complementary products, J. Revenue Pricing Manag. 21 (2) (2022) 164–182.

[31] E. Senyigit, B. Babayigit, Multi-criteria supplier selection and purchase problem, Sigma: J. Eng. Nat. Sci. 36 (4) (2018) 1253–1262.
[32] H. Schiele, E. Hofman, B.M. Zunk, J. Eggers, Why and how to involve purchasing in new product development? Int. J. Innov. Manag. 25 (3) (2021).
[33] S.A.F.S.A. Rahman, K.N.A. Maulud, B. Pradhan, S.N.A.S. Mustorpha, Manifestation of lattice topology data model for indoor navigation path based on the 3D building environment, J. Comput. Des. Eng. 8 (6) (2021) 1533–1547.
[34] K. Zhu, J. Wang, N. Guo, Z. Ding, P. Mei, Simulation of pedestrian counter flow with conflicting preference using a lattice-based simulation model, Internat. J. Modern Phys. C 31 (9) (2020).
[35] S.X. Duan, S. Wibowo, J. Chong, A multicriteria analysis approach for evaluating the performance of agriculture decision support systems for sustainable agribusiness, Mathematics 9 (8) (2021).
[36] Y. Fang, X. Zhang, Y. Li, Research on multicriteria decision-making scheme of high-speed railway express product pricing and slot allocation under competitive conditions, Mathematics 10 (9) (2022).
[37] A. Iampan, G.S. Garcia, M. Riaz, H.M.A. Farid, R. Chinram, Linear diophantine fuzzy Einstein aggregation operators for multi-criteria decision-making problems, J. Math. 2021 (2021).
[38] M. Jamil, F. Afzal, D. Afzal, D.K. Thapa, A. Maqbool, Multicriteria decision-making methods using bipolar neutrosophic Hamacher geometric aggregation operators, J. Funct. Spaces 2022 (2022).
[39] M. Juanpera, B. Domenech, L. Ferrer-Marti, A. Garcia-Villoria, R. Pastor, Methodology for integrated multicriteria decision-making with uncertainty: Extending the compromise ranking method for uncertain evaluation of alternatives, Fuzzy Sets and Systems 434 (2022) 135–158, Special Issue.
[40] W.C. Lo, C.H. Lu, Y.C. Chou, Application of multicriteria decision making and multi-objective planning methods for evaluating metropolitan parks in terms of budget and benefits, Mathematics 8 (8) (2020).
[41] L. Louvart, P. Meyer, A.-L. Olteanu, MODEL: A multicriteria ordinal evaluation tool for GIS, Int. J. Geogr. Inf. Sci. 29 (1910) (2015) 10–1931.
[42] D. Macek, I. Magdalenic, N.B. Redep, A model for the evaluation of critical IT systems using multicriteria decision-making with elements for risk assessment, Mathematics 9 (9) (2021).
[43] J. Muangman, K. Krootsong, P. Polrong, W. Yukunthorn, W. Udomsap, Fuzzy multicriteria decision-making for ranking intercrop in rubber plantations under social, economic, and environmental criteria, Adv. Fuzzy Syst. 2020 (2020).
[44] I. Chajda, H. Länger, Horizontal sums of bounded lattices, Math. Pannonica 20/1 (2009) 1–5.
[45] I. Chajda, H. Langer, Orthomodular lattices that are horizontal sums of Boolean algebras, Commentationes Math. Univ. Carolinae 61 (1) (2020) 11–20.
[46] J. Tkadlec, P. Zacek, States on effect algebras, their products and horizontal sums, Math. Slovaca 66 (5) (2016) 1029–1036.