Article

Asymmetrization of a Set of Degressively Proportional Allocations with Respect to Lexicographic Order. An Algorithmic Approach

Ewa Łyko ¹ , Janusz Łyko ², Arkadiusz Maciuk ² and Maciej Szczeciński ^{2,*}

¹ Faculty of Computer Science and Management, Wrocław University of Science and Technology, Wyspiarski St. 27, 50-370 Wrocław, Poland; ewa.lyko97@gmail.com

² Wrocław University of Economics and Business, Komandorska St. 118/120, 53-345 Wrocław, Poland; janusz.lyko@ue.wroc.pl (J.Ł.); arkadiusz.maciuk@ue.wroc.pl (A.M.)

* Correspondence: maciej.szczecinski@ue.wroc.pl

Abstract: In the case of the proportional allocation of goods and burdens, the shares of all agents with respect to their values are equal, i.e., they form a constant sequence. In a degressively proportional allocation this sequence is nondecreasing when agents are increasingly ordered according to their values. The division performed according to this principle is ambiguous, and its selection requires many negotiations among participants. The aim of this paper is to limit the range of such negotiations when the problem is complex, i.e., the set of feasible solutions has high cardinality. It can be done thanks to a numerical analysis of the set of all feasible solutions, and eliminating allocations favoring or disfavoring some coalitions of agents. The problem is illustrated by the case study of allocating seats in the European Parliament in its 2019–2024 term.

Keywords: asymmetrization; lexicographic order; fair division; degressively proportional allocation; European Parliament; sequential coalition

Citation: Łyko, E.; Łyko, J.; Maciuk, A.; Szczeciński, M. Asymmetrization of a Set of Degressively Proportional Allocations with Respect to Lexicographic Order. An Algorithmic Approach. *Symmetry* **2021**, *13*, 1269. <https://doi.org/10.3390/sym13071269>

Received: 2 June 2021

Accepted: 12 July 2021

Published: 15 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The problem of allocation of gains and burdens emerges in many aspects of social life. Local government entities and management boards of large corporations decide about fair distribution, while entire societies participate in general elections to allocate seats in collegial bodies to their representatives. In most cases the fair allocation depends on the values representing the respective participants in the division. Those values are sometimes called entitlements and are typically expressed by numbers. For example, when dividends are paid, the entitlements are specified by the numbers of shares held, and in the example of apportioning seats in the US House of Representatives to individual states, the entitlements are their populations.

A natural and widely applied principle of fair distribution is the rule of proportionality that originated from Aristotle. According to this principle, the share of each agent in the distributed good should be equal to the quotient of its value to the total value of all contenders. This principle functions well when the values of all agents differ moderately. When these differences are considerable, applying proportional division is likely to marginalize the contenders characterized by the smallest values. In practical cases, the common good of all agents implies that the contenders with the largest shares give up some part of their entitlements resulting from the proportionality principle for the sake of those agents whose values are smallest.

One such approach is degressively proportional allocation. In degressively proportional allocation, goods or burdens are divided among the participants of the division according to their entitlements, so that the higher entitlement agents receive several goods no lower than agents with lower entitlements. At the same time, their numbers are limited,

i.e., the ratio of the number of goods received to the entitlements of the larger entitlement agent must not exceed the ratio for the smaller entitlement agent. Given the sequence of contender values and a generally defined principle, a problem appears because there is no unique solution, as many different allocations satisfy the above-mentioned inequalities.

This paper presents a new approach to the problem of degressive proportionality. The authors do not try to indicate one concrete solution, but instead focus on a reduction of the set of all feasible allocations. Such an approach is motivated by assuming that when no legislative indications are available to perform the allocation, then a solution is chosen by means of negotiations among those participating in the division. This is the case of how the seats in the European Parliament are allocated.

So far there is no consensus as regards the application of a replicable rule to find the composition of this collegial body. The allocation of seats in successive terms has always been a result of political negotiations. In this context, it is important to know how many possible solutions are available to negotiators for their choice. The fewer the feasible solutions the better the review of possible variants and therefore, the more rational the choice.

To realize this idea, first of all, a LaRsa algorithm [1,2] was applied to find all possible allocations for the 2019–2024 term. The number of all possible allocations was more than 122 million.

Next, the criterion of searching the set of those allocations was specified as favoring or disfavoring a fixed set of agents by given allocation. Favoring and disfavoring are defined here in a similar way to the theory of cooperative games. One assumes that individual agents join a coalition in a fixed order, and they are allocated the maximal number of seats that is available at that time. This implies that a given allocation favors the ordered sequence—the permutation of agents—if it is the maximal element of all possible allocations which are lexicographically ordered with respect to this permutation.

A similar assumption is made for disfavoring—the only difference is that the agents obtain a minimal number of seats at that time. This implies that a given coalition disfavors some fixed permutation of agents if it is the minimal element of all possible allocations, ordered lexicographically with respect to this permutation.

In general, the problem can be brought down to an analysis of the set of all possible permutations. However, it turns out that the constraints resulting from the conditions of degressive proportionality allow this problem to be solved in a reasonable time frame. Implementing the algorithm presented in the paper, despite the high complexity of the problem, made it possible to find all maximal and minimal elements, and those permutations for which those elements are maximum (minimum) ones.

One may assume that such maximal and minimal elements are the hardest for negotiators to accept. Agents are not eager to accept a solution that is the least advantageous for them or that is the most advantageous for another agent. For this reason, by eliminating minimal and maximal elements for all permutations of agents from the set of all feasible solutions, the set under negotiation becomes reduced, with no harm to the choice of a desired solution. This reasoning leads to the exclusion of symmetrical allocations from all allocations, with respect to the order relation subsets, because $\max(X) = -\min(-X)$ for $X \subset \mathbb{R}^n$ in lexicographic order. The iterative application of this procedure possibly, with respect to order relation, allows the determination of the most asymmetric subset of the set of all allocations, which, from the point of view of negotiations, is the set of allocations most easily accepted by all participants of the division. In the case of the composition of the European Parliament for the 2019–2024 term the reduction would be tens of thousands of elements of all feasible solutions. It might not be very impressive, but one should remember that the procedure of eliminating maximal and minimal elements can be rerun in subsequent iterations of the proposed algorithm. In addition, some information was gained about favoring and disfavoring sequential coalitions, i.e., that the joining of another agent does not change, respectively, the maximal or minimal element for the entire permutation. As a result, it is possible to determine the best solution of the allocation

problem for each sequential coalition. In addition, each agent gains information about advantages and disadvantages of joining a given coalition in a given moment.

A similar procedure of the selection of the set under analysis into favoring and disfavoring elements and neutral elements by means of lexicographic orders has been used in papers from the areas of bargaining games [3,4], cooperative matching games [5] and sequential forward selection [6]. The method presented in this paper is different from the methods presented in the mentioned papers, because it does not require the objective function or other data besides the set of elements under analysis.

2. Problem Statement

In this section, we will formally define the analyzed problem. A nonnegative sequence of integers $S = (s_1, s_2, \dots, s_n)$ is called a degressively proportional allocation of h goods with respect to a positive and nondecreasing sequence of values $P = (p_1, p_2, \dots, p_n)$ if and only if for each $1 \leq i < n$: $s_i \leq s_{i+1}$, $\frac{s_i}{p_i} \geq \frac{s_{i+1}}{p_{i+1}}$, and $s_1 + s_2 + \dots + s_n = h$.

Let $h > 0$, $P = (p_1, p_2, \dots, p_n)$ be a positive sequence of values, $N = \{1, 2, \dots, n\}$ be the set of indices, $X = \{S = (s_1, s_2, \dots, s_n) \in \mathbb{R}^n\}$ be the m -elementary, $m > 0$, set of degressively proportional allocations of h goods with respect to P and $\Pi(N)$ the set of permutations of N . On set X one defines the class of relations of lexicographic orders determined by $\Pi(N)$. We will say that one feasible element from set X is not less than another element from this set in the relation of lexicographic order under a specific sequence of entities, if the vector of the first element is lexicographically not less than the vector of the second element with respect to this sequence.

Definition 1. (The lexicographic order regarding the permutation π)

$$S', S \in X, \pi \in \Pi(N). \quad S' \succ_{\pi} S \Leftrightarrow s'_{\pi(1)} > s_{\pi(1)} \vee \left(s'_{\pi(1)} = s_{\pi(1)} \wedge s'_{\pi(2)} > s_{\pi(2)} \right) \vee \dots \vee \left(s'_{\pi(1)} = s_{\pi(1)} \wedge s'_{\pi(2)} = s_{\pi(2)} \wedge \dots \wedge s'_{\pi(n-1)} > s_{\pi(n-1)} \right) \vee \left(s'_{\pi(1)} = s_{\pi(1)} \wedge s'_{\pi(2)} = s_{\pi(2)} \wedge \dots \wedge s'_{\pi(n-1)} = s_{\pi(n-1)} \wedge s'_{\pi(n)} \geq s_{\pi(n)} \right).$$

Lexicographic relation is a relation of linear order—any two elements of X are in relation. Hence each of these relations has got exactly one maximum and one minimum, i.e., the maximal (minimal) element, whereas one element can be the maximal (minimal) element for the relations determined by different permutations.

Definition 2. (Maximal and minimal elements with respect to given permutation) The maximal element for the relation π will be denoted by S_{π}^{max} and defined as follows

$$\forall S \in X \quad S \prec_{\pi} S_{\pi}^{max} \implies S_{\pi}^{max} \succ_{\pi} S.$$

The minimal element for the relation π will be denoted by S_{π}^{min} and defined as follows

$$\forall S \in X \quad S_{\pi}^{min} \prec_{\pi} S \implies S \succ_{\pi} S_{\pi}^{min}.$$

Definition 3. (The set of favoring, disfavoring and neutral elements with respect to the class of lexicographic relations).

$$S \in \text{Favor}(X) \Leftrightarrow \exists \pi \in \Pi(N) : S = S_{\pi}^{max};$$

$$S \in \text{Disfavor}(X) \Leftrightarrow \exists \pi \in \Pi(N) : S = S_{\pi}^{\min};$$

$$\text{Neutral}(X) = X \setminus (\text{Favor}(X) \cup \text{Disfavor}(X))$$

The set $\text{Neutral}(X)$ includes elements from set X , which are neither maximal nor minimal with respect to no permutation. During negotiations aimed at the selection of one division out of all feasible solutions, no agent will accept the allocation of the minimal number of goods available or the allocation of the maximal amount to another agent. In this case, $\text{Neutral}(X)$ can be interpreted as a set of allocations which does not favor or disfavor any sequential coalition of agents, so these allocations are more acceptable than elements of sets $\text{Favor}(X)$ and $\text{Disfavor}(X)$. The set $\text{Neutral}(X)$ can be a basis for efficient and reduced negotiations under transparent and equal conditions for all participants in division.

It should be emphasized that the sets $\text{Favor}(X)$ and $\text{Disfavor}(X)$ are symmetrical to each other: the equation $-\text{Favor}(X) = \text{Disfavor}(-X)$ holds, and therefore the determination of the set $\text{Neutral}(X)$ consists of reducing this set by the symmetrical subset $\text{Favor}(X) \cup -\text{Favor}(-X) = \text{Disfavor}(X) \cup -\text{Disfavor}(-X)$. Searching for a set of neutral elements can therefore be interpreted as a kind of asymmetrization of the set X . The iterative applying of the procedure described in the article, as mentioned earlier in the introduction, thus leads to reducing the set of all allocations to the most asymmetric set of allocations possible.

The method described in this paper was developed for applications with set X of degressively proportional divisions with lexicographic order. It has to be mentioned that it can also be, without any changes, applied when X is any subset of $\mathbb{R}^n$ with lexicographic order. Moreover, the algorithms presented in the paper can be easily modified for any order different than lexicographic. More precisely, it is enough to modify only four of them: SMAX, SMIN, LEXORDMAX and LEXORDMIN (wherein SMIN and LEXORDMIN are similar to SMAX and LEXORDMAX).

3. Methodology

Finding the set of maximal (minimal) elements by searching all possible permutations is complex and not efficient, because the number of all permutations is $n!$. To reduce the number of considered permutations, one may recursively construe subsequent subsets of set X formed by the vectors whose determined coordinate, in terms of a given sequence, has got the maximal (minimal) possible values. The more coordinates considered, the smaller the number of elements in subsequent sets. When the generated set is one element set, further iterations are unneeded. Thus, adding another agent to the generated arrangement of agents does not affect the maximal (minimal) element of the relation. Let us introduce the following notation. Let A be any subset of the set X , then

$$s_i^{\max(A)} = \max_{S \in A} \{s_i\}, \quad s_i^{\min(A)} = \min_{S \in A} \{s_i\}.$$

We can also define recursively family of sets

$$\begin{aligned} X^{\max}(\pi(1)) &= \left\{ S \in X : s_{\pi(1)} = s_{\pi(1)}^{\max(X)} \right\}, \dots, \\ &X^{\max}(\pi(1), \dots, \pi(k), \pi(k+1)) \\ &= \left\{ S \in X : s_{\pi(k+1)} = s_{\pi(k+1)}^{\max(X(\pi(1), \dots, \pi(k)))} \right\}. \end{aligned}$$

Of course, for every $k \leq n$, $X^{\max}(\pi(1), \dots, \pi(k)) \neq \emptyset$ and this family is descending. Analogically, we can define family of sets

$$\begin{aligned} X^{\min}(\pi(1)) &= \left\{ S \in X : s_{\pi(1)} = s_{\pi(1)}^{\min(X)} \right\}, \dots, \\ X^{\min}(\pi(1), \dots, \pi(k), \pi(k+1)) &= \left\{ S \in X : s_{\pi(k+1)} = s_{\pi(k+1)}^{\min(X(\pi(1), \dots, \pi(k)))} \right\}. \end{aligned}$$

In a similar way, for every $k \leq n$, $X^{\min}(\pi(1), \dots, \pi(k)) \neq \emptyset$ and this family is descending. As a result, Proposition 1 immediately follows.

Proposition 1. For $k \leq n$ such that $X^{\max}(\pi(1), \dots, \pi(k)) = \{S_{\pi}^{\max}\}$ we have $\{S_{\pi}^{\max}\} = X^{\max}(\pi(1), \dots, \pi(k)) = X^{\max}(\pi(1), \dots, \pi(k), \pi(k+1)) = \dots = X^{\max}(\pi(1), \dots, \pi(n))$.

Analogically, for $k \leq n$ such that $X^{\min}(\pi(1), \dots, \pi(k)) = \{S_{\pi}^{\min}\}$ we have $\{S_{\pi}^{\min}\} = X^{\min}(\pi(1), \dots, \pi(k)) = X^{\min}(\pi(1), \dots, \pi(k), \pi(k+1)) = \dots = X^{\min}(\pi(1), \dots, \pi(n))$.

Definition 4. (favoring and disfavoring sequential coalition). The ordered favoring (disfavoring) coalition of the permutation $\pi \in \Pi(N)$ is the shortest truncation $(\pi(1), \dots, \pi(k))$, $k \leq n$, of permutation π which determines a one-element set $Favor(X)$ ($Disfavor(X)$).

To generate the set of favoring (disfavoring) elements, it suffices to iteratively divide the set X , concluding with one-element sets, and then to take their sum. By assigning truncated permutations to maximal (minimal) elements, the identification of favoring and disfavoring sequential coalitions is achievable. The elements which are truncated in this procedure form the set of neutral elements.

The idea of the proposed algorithm to split the set X into the set of favoring, disfavoring and neutral elements, which can be illustrated by the following example. Table 1 presents the set $X = \{S_1, S_2, \dots, S_{11}\}$ with maximas for every coordinate in every vector (marked by overlines) and minima (marked by underlines). The subsets $X^{\max}(1) = \{S_{11}\}$, $X^{\max}(2) = \{S_5\}$, $X^{\max}(3) = \{S_1\}$, $X^{\max}(4) = \{S_1, S_2, S_3, S_6, S_8\}$, $X^{\max}(5) = X$ are a result of splitting for maximal elements in regard to one coordinate. S_{11} is the maximal element for the orders determined by permutations starting with 1, S_5 for the orders determined by permutations starting with 2 and S_1 for the orders determined by permutations starting with 3. In this case, considering the orders determined by permutation starting with 5 is redundant, as the set $X^{\max}(5, 1)$ will be equal to the set $X^{\max}(1)$, $X^{\max}(5, 2)$ to the set $X^{\max}(2)$ and so on. The sets $X^{\max}(4, 1) = \{S_6, S_8\}$, $X^{\max}(4, 2) = \{S_3, S_8\}$ and $X^{\max}(4, 3) = \{S_1\}$ remain to be considered, and in the third iteration we get $X^{\max}(4, 1, 2) = \{S_8\}$, $X^{\max}(4, 1, 3) = \{S_6\}$, $X^{\max}(4, 2, 1) = \{S_8\}$ and $X^{\max}(4, 2, 3) = \{S_3\}$. The set of favoring elements is the sum of those one-element sets, i.e., $Favor(X) = \{S_1, S_3, S_5, S_6, S_8, S_{11}\}$. In the same way, we can obtain set $Disfavor(X) = \{S_1, S_3, S_4, S_5, S_8, S_{10}, S_{11}\}$.

Table 1. Sample set of vectors with maximas and minimas.

	P	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
1	90	<u>15</u>	<u>15</u>	<u>15</u>	<u>15</u>	<u>15</u>	16	16	16	16	17	<u>18</u>
2	65	<u>11</u>	<u>12</u>	<u>13</u>	<u>13</u>	<u>14</u>	12	12	13	13	13	13
3	40	<u>11</u>	10	9	10	9	9	10	<u>8</u>	9	<u>8</u>	<u>8</u>
4	20	<u>6</u>	<u>6</u>	<u>6</u>	5	5	<u>6</u>	5	<u>6</u>	5	5	<u>4</u>
5	10	3	3	3	3	3	3	3	3	3	3	3

The set of neutral elements is formed by elements which are not assigned to none of these sets, i.e., $Neutral(X) = \{S_2, S_7, S_9\}$, which results in a significant decrease of the number of arithmetic operations in subsequent operations. The favoring sequential coalitions disregarding the fifth coordinate are: $\{(1), (2), (3), (4, 1, 2), (4, 1, 3), (4, 2, 1), (4, 2, 3), (4, 3)\}$. The disfavoring sequential coalitions disregarding the fifth coordinate are: $\{(2), (4), (1, 2), (3, 4), (3, 1), (1, 3, 2), (1, 3, 4), (1, 4, 2), (1, 4, 3), (3, 2, 1), (3, 2, 4)\}$. Considering this coordinate in practice doubles the set of both types of critical coalitions.

4. Algorithms

This section describes the algorithms which determine the set of favoring and disfavoring elements and their corresponding truncated permutations. It also describes an algorithm which determines a set of neutral elements. So as to handle the potential size of the input set under analysis, these algorithms employ the concept of stack and a map

in the form of a rooted tree graph. The main algorithms are Algorithm 1 (whose components are Algorithms 2 and 3) for favoring elements, Algorithm 4 (whose components are Algorithms 5 and 6) for disfavoring elements and Algorithm 7 (whose component is Algorithm 8 and also uses the results of Algorithms 1 and 4) for neutral elements. It can be seen that Algorithms 1 and 4 work symmetrically: the procedure for finding the minimal, with respect to lexicographic order, elements of the set X of all allocations is identical to the procedure for finding its maximal elements, except that the inequality in the corresponding conditional statements changes. Moreover, equality $-Favor(X) = Disfavor(-X)$ holds, so multiplying all the elements of the set of all allocations by -1 makes the result of Algorithm 6 for the set $-X$ the same as result of Algorithm 3 for X , but with the opposite sign. Complete descriptions of the algorithms are presented in Appendix A.

In the whole of Section 4 and also in description of the algorithms, we will use the following notation. Let $N = \{1, 2, \dots, n\}$ as previously be the set of indices and let $\pi \in \Pi(N)$ be some permutation of N . By σ we denote a truncation of permutation π and by σi_k we denote extension of σ by index $i_k \in N$ which is not an element of σ . Moreover, for every σ we denote by $X(\sigma)$ the set of all elements favoring (in algorithm LEXORDMAX) or disfavoring (in algorithm LEXORDMIN) truncated permutation σ .

Algorithm 1. Algorithm for Determining Lexicographically Favoring Allocations FAVOR(X, N).

Input: The stack of vectors X , the set N of indices of players.

Output: Set F , whose elements are sets $\{s, \sigma_1, \sigma_2, \dots, \sigma_{k(s)}\}$, where s is a favoring element, and $\sigma_1, \sigma_2, \dots, \sigma_{k(s)}$ are critical coalitions favored by it. The stack of not favoring elements NF .

1. If $\overline{X} = 1$, then remove all the elements from the stack X onto the stack B and STOP. Otherwise, go to 2.
 2. Create the graph $G(V, E)$, where $V = \{X\}$, $E = \phi$.
 3. Perform the algorithm LEXORDMAX(ϕ, X, N). Add vertices corresponding with triplets: $\{X(1), L_1, 1\}, \dots, \{X(\overline{N}), L_{\overline{N}}, \overline{N}\}$ to the graph G . Remove the elements from the stack K onto the stack NF . Join the vertices with vertex of X by edges. Record the pairs in physical memory and delete from the cache memory.
 4. Locate in the vertex X .
 5. If there are edges downwards, then go to 6. Otherwise go to 9.
 6. Select one edge downwards and go to the next vertex.
 7. Read the triplet $\{X(\sigma), L, \sigma\}$ corresponding with the current vertex. If $L = 1$, then go to 8. Otherwise go to 9.
 8. If the element from the stack $X(\sigma)$ is the element of one of the sets in the set F , add σ to this set. Otherwise add the set $\{X(\sigma), \sigma\}$ to the set F . Go upwards along the edge, delete the current vertex and its incidental edge. Go to 5.
 9. Perform the algorithm LEXORDMAX($\sigma, X(\sigma), N$). Add vertices corresponding with the triplets: $\{X(\sigma i_1), L_1, \sigma i_1\}, \dots, \{X(\sigma i_{\overline{N(\sigma)}}), L_{\overline{N}}, \sigma i_{\overline{N(\sigma)}}\}$ to the graph G . Remove the elements of the stack K onto the stack NF . Join the vertices with the vertex $\{X(\sigma), L, \sigma\}$ by the edges. Record the pairs in the physical memory and delete from the cache memory. Go to 5.
 10. If the current vertex is X , then STOP. Otherwise go upwards along the edge and delete the current vertex and its incidental edge. Go to 5.
-

This algorithm returns a set of all elements from X that favor at least one coalition and sets of coalitions favored by individual elements. The output from the LEXORDMAX algorithm is employed here.

Algorithm 2. Algorithm Determining the Maximal Elements on Each Coordinate of SMAX(X).**Input:** The stack of vectors X .**Output:** The vector SMAX(X), whose each coordinate equals the maximal value among all the vectors from the stack X .

1. Remove the vector v from the stack X .
2. Let SMAX(X) = v .
3. If the stack X is empty, then STOP. Otherwise remove the vector v from the stack X .
4. Compare the values on subsequent coordinates of the vector v with the corresponding values of the vector SMAX(X). If any value from the vector v is greater, substitute the value of the respective coordinate of the vector SMAX(X) by this value. Go to 3.

The SMAX algorithm returns the maximum values over each coordinate of all vectors from the stack X and the rearranged initial stack X .

Algorithm 3. Algorithm Splitting a Set into Smaller Subsets according to the Lexicographic Order LEXORDMAX($\sigma, X(\sigma), N$).

Given a truncated permutation σ , LEXORDMAX algorithm divides the stack $X(\sigma)$ into (not necessarily disjointed) stacks of elements, which favor the permutations $\sigma i_1, \dots, \sigma i_k$, where $i_1, \dots, i_k$ are indexes from the set N , which are not the elements of the permutation σ . The algorithm returns triplets of the form $\{X(\sigma i_q), L_q, \sigma i_q\}$, where $X(\sigma i_q)$ is a stack of elements favoring the permutation σi_q and L_q is its cardinality. This algorithm employs the output from the SMAX algorithm.

Algorithm 4. Algorithm of Determining Lexicographically Disfavoring Allocations DISFAVOR(X, N).**Input:** The stack of vectors X , the set N of indices of players.**Output:** The set F , whose elements are sets $\{s, \sigma_1, \sigma_2, \dots, \sigma_{k(s)}\}$, where s is a favoring element, and $\sigma_1, \sigma_2, \dots, \sigma_{k(s)}$ are critical coalitions favored by it. The stack of not disfavoring elements of NDF .**Description of algorithm:** This algorithm works similarly to algorithm FAVOR(X, N). The differences are in step 3 and step 9, where algorithm DISFAVOR(X, N) performs algorithm LEXORDMIN (instead of LEXORDMAX).

This algorithm returns a set of all elements from X disfavoring at least one coalition and sets of coalitions disfavored by individual elements. The output from the LEXORDMIN algorithm is employed here.

Algorithm 5. Algorithm Determining the Maximal Elements on Each Coordinate of SMIN(X).**Input:** The stack of vectors X .**Output:** The vector SMIN(X), whose each coordinate equals the minimal value among all the vectors from the stack X .**Description of algorithm:** This algorithm works very similarly to algorithm SMAX(X). The only difference is step 4, where algorithm SMIN(X) substitutes coordinates of vector SMIN(X) with values from vector v , which are smaller than values of SMAX(X).

The SMIN algorithm returns minimal values over each coordinate among all vector from the stack X .

Algorithm 6. Algorithm Splitting a Set into Smaller Subsets according to the Lexicographic Order LEXORDMIN($\sigma, X(\sigma), N$).

Input: The stack of vectors $X(\sigma)$, set of indices N , ordered subset of indices σ .

Output: Stacks, their corresponding sizes and truncated permutations $\{X(\sigma_{i_1}), L_1, \sigma_{i_1}\}, \dots, \{X(\sigma_{i_k}), L_k, \sigma_{i_k}\}$. σ_{i_k} is formed by adding the index i_k at the end of the set σ . The stack K .

Description of algorithm: This algorithm works similarly to algorithm LEXORDMAX($\sigma, X(\sigma), N$). The differences are in step 1 and step 5. In these steps algorithm LEXORDMIN($\sigma, X(\sigma), N$) performs algorithm SMIN($X(\sigma)$) (instead of SMAX($X(\sigma)$)). Moreover, in step 5 vector v is added to the stack K if values on all coordinates from set $N(\sigma)$ are greater than values on these coordinates of vector SMIN($X(\sigma)$).

Given a truncated permutation σ LEXORDMIN algorithm divides the stack $X(\sigma)$ into (not necessarily disjointed) stacks of elements, which disfavor the permutations $\sigma_{i_1}, \dots, \sigma_{i_k}$, where $i_1, \dots, i_k$ are indexes from the set N , which are not elements of permutation σ . The algorithm returns triplets of the form $\{X(\sigma_{i_q}), L_q, \sigma_{i_q}\}$, where $X(\sigma_{i_q})$ is a stack of elements disfavoring the permutation σ_{i_q} and L_q is its cardinality. This algorithm employs the output from the SMIN algorithm.

Algorithm 7. Algorithm of Determining Lexicographically Neutral Allocations NEUTRAL(X, N).

Input: The stack of vectors X , the set N of indices of players.

Output: The stack of neutral elements *NEUTRAL*.

1. Perform algorithm FAVOR(X, N). Save only stack *NF*.
 2. Perform algorithm DISFAVOR(X, N). Save only stack *NDF*.
 3. If stack *NF* is non-empty, take vector v from *NF* and go to 4. Otherwise, go to 5.
 4. Perform algorithm ISIN(v, NDF). If result is 1, put v on stack *NEUTRAL*. Otherwise, remove v . Go to 3.
 5. STOP.
-

The NEUTRAL algorithm returns a stack of neutral elements from stack X . Running this algorithm requires outputs from all the algorithms described above.

Algorithm 8. Algorithm to Check if Vector v Is an Element of Stack S ISIN(v, S).

Input: Vector v , the stack of vectors S .

Output: Logical value *Is*.

1. Substitute $Is = 0$.
2. If stack S is non-empty, take vector s from stack S . Otherwise, STOP.
3. If $v = s$, substitute $Is = 1$. STOP. Otherwise, remove s and go to 2.

The ISIN algorithm checks whether vector v is an element from the stack S .

Let m denote cardinality of the set of vectors, n the number of coordinates of a single vector from the set of vectors. Let $E_{NEUTRAL}$ denote the number of elementary operations performed by the NEUTRAL(X, N) algorithm. The following proposition holds (the proof is presented in Appendix A).

Proposition 2. $E_{NEUTRAL}$ satisfies in worst case inequalities

$$\begin{aligned} 3 + 2 \sum_{p=1}^{n-2} \left(m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} \right) + 4n(n-1)! &\leq E_{NEUTRAL} \\ &\leq 3 + 4 \sum_{p=1}^{n-1} \left(\prod_{j=1}^p (n-j) \right) \\ &\quad + 2 \sum_{p=1}^{n-2} \left(2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \right) \\ &\quad + 2(n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right). \end{aligned}$$

5. Case Study

The increased interest in degressively proportional allocation, and thereby the growth of research concerning it, has been occurring since 2007, when the Treaty of Lisbon introduced the principle as a basis of allocating mandates in the European Parliament among the member states [7–13]. In it, the populations of the respective countries are assumed as their values or entitlements. Moreover, in addition to the total number of seats in the European Parliament, which is a natural element of any allocation, this act of law also specifies the so-called boundary conditions of allocation, i.e., the minimal and maximal numbers of seats which can be allocated to the most and least populated countries in the European Union. In this variant, the problem is reduced to only integer allocations under given boundary conditions. Nevertheless, there may be many feasible solutions.

The research into degressive proportionality is markedly carried out in two threads. The first one draws on standard solutions applied in cases of proportional integer allocations [14]. One assumes that the condition of degressive proportionality has to be met before merely rounding to integers [15–19]. The second thread builds upon approaching the rule in its literal wording, and thereby it is assumed that the sequence which performs a given allocation must be degressively proportional [20–23]. In this case, a numerical analysis of the set of all feasible solutions plays an important role. The paper [1] introduces an algorithm, LaRsa, that allows the generation of such a set for the problem of allocating the seats in the European Parliament. Therefore, the complete analysis of the problem and the choice of one solution according to the predefined criterion are enabled (see also [2]).

The algorithm presented in Section 4 was applied with degressively proportional allocations, where $P = (434403, 576249, 848319, 1315944, 1968957, 2064188, 2888558, 4190669, 4664156, 5407910, 5465408, 5700917, 7153784, 8711500, 9830485, 9998000, 10341330, 10445783, 10793526, 11289853, 17235349, 19759968, 37967209, 46438422, 61302519, 66661621, 82064489)$. The data on the populations of the member states (without the United Kingdom), i.e., the terms of the sequence $P = (p_1, p_2, \dots, p_n)$ were acquired from the Eurostat, while the minimal and maximal numbers of seats, that can be assigned to particular country are given by the Treaty of Lisbon, and are equal to 6 and 96, respectively. Furthermore, after Brexit, $h = 705$.

The implementation of the algorithm presented in this paper allowed, despite the high computational complexity, the determination of all maximal elements and their permutations. Thus, defining the best solution is possible for each sequential coalition. Furthermore, each agent gains information about the gains from joining a given coalition in a certain moment, as well as about sequential critical coalitions.

The result was 2455 favoring and 19,786 disfavoring elements in the set of all allocations. Corresponding to these allocations were 42,613 favoring sequential coalitions and 269,023,606 disfavoring sequential coalitions.

Below we present an additional analysis of the set of all favoring and disfavoring sequential coalitions for permutations that did not start from Luxembourg, from Italy or from France. The reason for omitting these coalitions is the number of these coalitions, which causes high complexity in the problem. For example, there were 213,968,740 disfavoring se-

quential coalitions started from Italy, France and Luxembourg. The analysis of such a large number of sequential coalitions, unfortunately, exceeds our current hardware capabilities.

In every division (favoring, disfavoring or neutral), Slovenia received the same number of seats as Latvia. Similarly, the same amount of goods were received by the set consisting of Slovakia, Denmark and Finland and the set consisting of Greece, Czech Republic, Portugal, Sweden and Hungary. The set of all allocations of parliamentary seats was thus insignificantly reduced (by less than 1%). However, it should be noted that the algorithms presented in this paper, if run many times, significantly reduce the set of all feasible allocations.

The number of divisions favoring and disfavoring at least one n -element critical coalition (due to n) is presented in Table 2.

Table 2. Number of divisions favoring and disfavoring at least one n -element sequential coalition.

n	Number of Favoring Divisions	Number of Disfavoring Divisions	n	Number of Favoring Divisions	Number of Disfavoring Divisions
1	1	0	9	4710	10,392,262
2	55	23	10	4710	13,337,745
3	440	1058	11	52	11,947,155
4	1228	11,540	12	12	7,177,959
5	1499	100,883	13		2,607,415
6	1066	597,102	14		947,081
7	527	2,215,739	15		14,700
8	209	5,704,204			

The smallest favoring sequential coalition in the analyzed set was formed by one country. There was only one such coalition, formed by Romania. There were no coalitions disfavoring only one country. The smallest disfavoring sequential coalition was formed by two countries, and there were 23 coalitions of this type. The largest favoring sequential coalitions were formed by twelve member states. There were 12 coalitions of this type. Luxembourg, Cyprus, Estonia and Ireland were present in all of them. The largest disfavoring coalitions were bigger: every one of them was formed by fifteen member states. There were 14,700 coalitions of this size. The most frequent favoring sequential coalitions were formed by nine and ten countries, and the most frequent disfavoring sequential coalition was formed by ten countries. There were 4710 favoring coalitions consisting of nine and 4710 favoring coalitions consisting of ten countries and 13,337,745 disfavoring coalitions consisting of ten countries. Moreover, one can notice that the distribution of the number of disfavoring coalitions with respect to size is symmetrical (with a 10-member coalition at the middle), while same distribution for favoring coalitions is strongly left asymmetrical.

Table 3 shows the number of favoring and disfavoring sequential coalitions in which each country participated. Estonia participated in the greatest number of favoring sequential coalitions. This country was a member of 26,526 favoring sequential coalitions. Cyprus participated in the greatest number of disfavoring coalitions, which was 42,013,166. Slovakia, Finland and Denmark were countries which participated in the smallest number of favoring sequential coalitions. Each of these countries was a member of 1739 such coalitions. The countries which participated in the lowest number of disfavoring coalitions—which was equal to 7,891,381—were Hungary, Sweden, Portugal, Czech Republic and Greece.

Table 3. Number of favoring and disfavoring sequential coalitions by country.

Country	Number of Favoring Sequential Coalitions	Number of Disfavoring Sequential Coalitions	Country	Number of Favoring Sequential Coalitions	Number of Disfavoring Sequential Coalitions
LU	24,777	20,984,967	HU	4710	7,891,381
CY	18,921	42,013,166	SE	4710	7,891,381
EE	26,526	34,978,460	PT	4710	7,891,381
LV	5741	16,572,878	CZ	4710	7,891,381
SI	5471	16,572,878	EL	4710	7,891,381
LT	11,817	28,766,675	BE	6158	9,488,761
HR	18,148	37,391,354	NL	12,378	23,264,545
IE	20,029	39,003,346	RO	7190	28,332,736
SK	1739	19,399,546	PL	14,102	31,642,673
FI	1739	19,399,546	ES	11,788	28,383,230
DK	1739	19,399,546	IT	8287	16,513,761
BG	17,093	35,592,655	FR	4473	19,569,784
AT	18,380	32,394,097			

Column “Country” shows the official abbreviations used in the European Union; LU stands for Luxembourg, CY—Cyprus, EE—Estonia, LV—Latvia, SI—Slovenia, LT—Lithuania, HR—Croatia, IE—Ireland, SK—Slovakia, FI—Finland, DK—Denmark, BG—Bulgaria, AT—Austria, HU—Hungary, SE—Sweden, PT—Portugal, CZ—the Czech Republic, EL—Greece, BE—Belgium, NL—the Netherlands, RO—Romania, PL—Poland, ES—Spain, IT—Italy and FR—France.

Since the number of coalitions favored or disfavored by single division changes from one to 1,703,731, we do not present table with data used in analysis below.

There are five favoring divisions of 630 coalitions that is the greatest number of coalitions favored by a single division. Furthermore, 325 divisions favor just one coalition. The greatest number of sequential coalitions disfavored by one division was 1,703,731—just one division disfavored such several coalitions. Similarly, just one division disfavored one sequential coalition. The most frequent number of coalitions that were disfavored by one division was 7—there were 840 divisions with this property.

6. Conclusions

Due to the potentially high cardinality of the set of acceptable (feasible) solutions, the use of the degressive proportionality principle could be a complex problem. So far, none of the allocation rules developed on its basis has gained universal acceptance, and therefore, in every case of its application, such as the allocation of seats in European Parliament, allocation is made through negotiations. If one assumes that the preferences of n participants of the allocation are represented by their lexicographic ordering, then the number of variants to be considered is equal to $n!$. The implementation of the algorithms presented in the paper allows, thanks to the sequential elimination of solutions that are difficult to accept in negotiations, for a significant reduction of the allocation variants necessary to be considered, making the negotiations more effective. Taking advantage of the fact that in the case of the allocation of seats in the European Parliament, in addition to the assumption of degressive proportionality, the boundary conditions of this allocation are also defined, it is possible to reasonably limit the computation time measured by the number of elementary operations. As a result, negotiations aimed at obtaining an allocation under fully transparent and equal circumstances are possible.

The big advantage of the described approach is the possibility of applying it, without any changes, for any set of allocations with lexicographic order, not necessarily degressively proportional. Moreover, after modifying four (in practice, two) algorithms, it is also possible to reduce the set of allocations with any order other than lexicographic.

Moreover, in the problem of allocating seats to the European Parliament, the implementation of the presented algorithms allows, in a reasonable time and without the use of supercomputers, computing clusters, etc., the determination of all favoring and almost all disfavouring sequential coalitions. In practice, this means that a participant in the allocation

receives information about the effect of joining an existing sequential coalition has on the allocation of seats.

Moreover, the number of favoring and disfavoring sequential coalitions containing given participant of the allocation, proves its importance among all participants, because remaining outside such coalitions means no influence on the allocation made. In the analyzed case, the mentioned sequential coalitions are not very large. Only in two cases did they exceed half of the total number of participants in the allocation. It follows that in this case, almost always less than half of the participants determine the entire allocation. Because of this, in the perspective of forthcoming research, analysis of allocations which favor or disfavor the sequential coalitions of agents makes it possible and effective to approach the problem of degressively proportional allocation as an issue that can be considered as a concept similar to the concept of Shapley values known from the area of claims problems.

Author Contributions: Conceptualization, J.Ł. and A.M.; Data curation, E.Ł.; Formal analysis, E.Ł. and M.S.; Methodology, E.Ł., J.Ł., A.M. and M.S.; Project administration, E.Ł.; Supervision, J.Ł.; Writing—original draft, J.Ł. and M.S.; Writing—review & editing, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: The project was financed by the Ministry of Science and Higher Education in Poland under the programme “Regional Initiative of Excellence” 2019–2022 project number 015/RID/2018/19 total funding amount 10 721 040,00 PLN”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://ec.europa.eu/eurostat/web/population-demography/demography-population-stock-balance/database> accessed on 14 July 2021.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Proposition 2

Lemma A1. Let E_{SMAX} and E_{SMIN} denote the number of elementary operations performed by algorithms $SMAX(X)$ and $SMIN(X)$, respectively. Then

$$E_{SMAX} = E_{SMIN} = n\overline{\overline{X}},$$

where $\overline{\overline{X}}$ denotes the cardinality of X .

Proof of Lemma A1. $SMAX(X)$ algorithm compares all the coordinates of each vector from the stack X with the values of respective coordinates of the vector $SMAX(X)$, so this algorithm performs $n\overline{\overline{X}}$ comparisons. If we treat every comparison as an elementary operation, the algorithm $SMAX(X)$ needs $n\overline{\overline{X}}$ elementary operations to obtain its result. The same equality is analogously proven for E_{SMIN} . $\square$

Lemma A2. Let $E_{LEXORDMAX}$ and $E_{LEXORDMIN}$ denote the numbers of elementary operations performed by $LEXORDMAX(\sigma, X(\sigma), N)$ and $LEXORDMIN(\sigma, X(\sigma), N)$ algorithms, respectively. Then the inequalities hold:

$$\begin{aligned} \overline{\overline{X(\sigma)}} + n\overline{\overline{X(\sigma)}} + \overline{\overline{\sigma}} &\leq E_{LEXORDMAX} \leq 2n\overline{\overline{X(\sigma)}} - \frac{1}{6}\overline{\overline{\sigma}}(\overline{\overline{\sigma}} + 1)(-3n + \overline{\overline{\sigma}} - 1), \\ \overline{\overline{X(\sigma)}} + n\overline{\overline{X(\sigma)}} + \overline{\overline{\sigma}} &\leq E_{LEXORDMIN} \\ &\leq 2n\overline{\overline{X(\sigma)}} - \frac{1}{6}\overline{\overline{\sigma}}(\overline{\overline{\sigma}} + 1)(-3n + \overline{\overline{\sigma}} - 1). \end{aligned}$$

Proof of Lemma A2. The largest number of coordinates in set $N(\sigma)$ is n and smallest number is 1. The number of comparisons for every vector from stack X performed by algorithm LEXORDMAX is $\overline{N(\sigma)}$, so it is between 1 and n . Because of this, the total number of comparisons done by algorithm LEXORDMAX is equal to $\overline{N(\sigma)}\overline{X(\sigma)}$ which is between $\overline{X(\sigma)}$ and $n\overline{X(\sigma)}$. Moreover, let us note that algorithm LEXORDMAX performs in the first step algorithm SMAX $X(\sigma)$, whose minimal number of elementary operations is $\overline{X(\sigma)}$. Now let us note that LEXORDMAX in second step obtain set $N(\sigma)$. In most optimistic case, every time first element taken from N is equal to first element taken from σ , so it needs to do $\overline{\sigma}$ comparisons. In the most pessimistic case, every time, the last element taken from N is equal to the last element taken from σ , so it needs to do $\sum_{i=0}^{\overline{\sigma}} (n-i)(\overline{\sigma}-i) = -\frac{1}{6}\overline{\sigma}(\overline{\sigma}+1)(-3n+\overline{\sigma}-1)$. The total number of elementary operations performed by algorithm LEXORDMAX is therefore between $\overline{X(\sigma)} + n\overline{X(\sigma)} + \overline{\sigma}$ and $n\overline{X(\sigma)} + n\overline{X(\sigma)} - \frac{1}{6}\overline{\sigma}(\overline{\sigma}+1)(-3n+\overline{\sigma}-1)$. If we denote it by $E_{\text{LEXORDMAX}}$, then

$$\overline{X(\sigma)} + n\overline{X(\sigma)} + \overline{\sigma} \leq E_{\text{LEXORDMAX}} \leq 2n\overline{X(\sigma)} - \frac{1}{6}\overline{\sigma}(\overline{\sigma}+1)(-3n+\overline{\sigma}-1).$$

Analogously we prove that

$$\overline{X(\sigma)} + n\overline{X(\sigma)} + \overline{\sigma} \leq E_{\text{LEXORDMIN}} \leq 2n\overline{X(\sigma)} - \frac{1}{6}\overline{\sigma}(\overline{\sigma}+1)(-3n+\overline{\sigma}-1).$$

□

Lemma A3. Let E_{FAVOR} and E_{DISFAVOR} denote the number of elementary operations performed by FAVOR(X, N) and DISFAVOR(X, N) algorithms, respectively. Then the inequalities hold

$$\begin{aligned} 1 + \sum_{p=1}^{n-2} \left(m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} \right) + 2n(n-1)! &\leq E_{\text{FAVOR}} \\ &\leq 1 + 2 \sum_{p=1}^{n-1} \left(\prod_{j=1}^p (n-j) \right) \\ &\quad + \sum_{p=1}^{n-2} \left(2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \right) \\ &\quad + (n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right), \end{aligned} \quad (\text{A1})$$

$$\begin{aligned} 1 + \sum_{p=1}^{n-2} \left(m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} \right) + 2n(n-1)! &\leq E_{\text{DISFAVOR}} \\ &\leq 1 + 2 \sum_{p=1}^{n-1} \left(\prod_{j=1}^p (n-j) \right) \\ &\quad + \sum_{p=1}^{n-2} \left(2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \right) \\ &\quad + (n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right). \end{aligned} \quad (\text{A2})$$

Proof of Lemma A3. Let us note that in the worst case, if we did not delete the vertices in the graph G , then its construction would be as follows:

Root X .

$V_1 = n - 1$ edges passing from the root, therefore $n - 1$ vertices on the first level.

$V_2 = (n - 1)(n - 2)$ edges on the second level (because $n - 2$ edges pass from each of $n - 1$ vertices on the first level).

$V_3 = (n - 1)(n - 2)(n - 3)$ vertices on the third level.

$V_p = (n-1)(n-2)(n-3) \dots (n-p)$ vertices on the p th level.

$V_{n-1} = (n-1)(n-2)(n-3) \dots 2 \cdot 1 = (n-1)!$ vertices on the last, $(n-1)$ th level.

Let us note that we perform LEXORDMAX($\sigma, X(\sigma), N$) algorithm on every vertex. In the worst-case scenario, on every level, except the last one, set X will not be reduced (e.g., the sum of sets $X(\sigma)$ is equal to X). Moreover, let us notice that on the q th level of sets $X(\sigma)$ there is V_{j-1} (implied by the above-described construction of graph G), where $V_0 = 1$. Let $\overline{\overline{X}}_q(\sigma)$ denote the cardinality of the set on the q th level; then we get $\overline{\overline{X}}_q(\sigma) = \frac{\overline{\overline{X}}}{V_{j-1}} = \frac{m}{V_{j-1}}$ for $q = 1, \dots, n-2$ and $\overline{\overline{X}}_{n-1}(\sigma) = 1$. Moreover, the differences between values $\overline{\overline{N}}(\sigma)$ (and also $\overline{\overline{\sigma}}$) on each subsequent two levels are 1 and $\overline{\overline{N}}(\sigma) = n-1$, $\overline{\overline{\sigma}} = 1$ on the first level. Therefore, the following structure of values $\overline{\overline{X}}(\sigma)$, $\overline{\overline{N}}(\sigma)$ and $\overline{\overline{\sigma}}$ on each level holds.

$$\overline{\overline{X}}_1(\sigma_1) = \frac{m}{V_0} = \frac{m}{1}, \overline{\overline{N}}_1(\sigma_1) = n-1, \overline{\overline{\sigma}}_1 = 1;$$

$$\overline{\overline{X}}_2(\sigma_2) = \frac{m}{V_1} = \frac{m}{(n-1)}, \overline{\overline{N}}_2(\sigma_2) = n-2, \overline{\overline{\sigma}}_2 = 2;$$

$$\overline{\overline{X}}_p(\sigma_p) = \frac{m}{V_{p-1}} = \frac{m}{(n-1)(n-2) \dots (n-p+1)} = \frac{m}{\prod_{i=1}^{p-1} (n-i)}, \overline{\overline{N}}_p(\sigma_p) = n-p, \overline{\overline{\sigma}}_p = p, \text{ where } p < n-1;$$

$$\overline{\overline{X}}_{n-1}(\sigma_{n-1}) = 1, \overline{\overline{N}}_{n-1}(\sigma_{n-1}) = 1, \sigma_{n-1} = n-1.$$

LEXORDMAX($\sigma, X(\sigma)$) algorithm is performed V_p times on the p th level; hence, the number of elementary operations performed exclusively by LEXORDMAX algorithm (implemented in FAVOR) on the p th level of graph G , denoted by $E_{\text{LEXORDMAX},p}$, is expressed by inequalities

$$\begin{aligned} V_{p-1}(\overline{\overline{X}}(\sigma_p) + n\overline{\overline{X}}(\sigma_p) + \overline{\overline{\sigma}}_p) &\leq E_{\text{LEXORDMAX},p} \\ &\leq V_{p-1} \left(2n\overline{\overline{X}}(\sigma_p) - \frac{1}{6}\overline{\overline{\sigma}}_p(\overline{\overline{\sigma}}_p + 1)(-3n + \overline{\overline{\sigma}}_p - 1) \right) \end{aligned}$$

which yields after some transformations

$$\begin{aligned} V_{p-1} \left(\frac{m}{V_{p-1}} + n\frac{m}{V_{p-1}} + p \right) &\leq E_{\text{LEXORDMAX},p} \\ &\leq V_{p-1} \left(2n\frac{m}{V_{p-1}} - \frac{1}{6}p(p+1)(-3n + p - 1) \right), \\ m + n + \frac{p}{V_{p-1}} &\leq E_{\text{LEXORDMAX},p} \leq 2mn + V_{p-1} \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right), \\ m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} &\leq E_{\text{LEXORDMAX},p} \\ &\leq 2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \end{aligned}$$

for $p < n-1$, as well as

$$\begin{aligned} (n-1)!(1+n + n-1) &\leq E_{\text{LEXORDMAX},n-1} \\ &\leq (n-1)! \left(2n - \frac{1}{6}(n-1)((n-1)+1)(-3n + (n-1) - 1) \right), \end{aligned}$$

$$2n(n-1)! \leq E_{\text{LEXORDMAX},n-1} \leq (n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right).$$

The total number of elementary operations performed by LEXORDMAX in the worst-case scenario of this algorithm is thus equal to at least

$$\sum_{p=1}^{n-2} \left(m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} \right) + 2n(n-1)!$$

and not exceeding

$$\sum_{p=1}^{n-2} \left(2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \right) + (n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right).$$

We therefore have

$$\begin{aligned} \sum_{p=1}^{n-2} \left(m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} \right) + 2n(n-1)! &\leq E_{\text{LEXORDMAX}} \\ &\leq \sum_{p=1}^{n-2} \left(2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \right) \\ &\quad + (n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right). \end{aligned}$$

In the worst-case scenario the number of operations of adding vertices equals $1 + \sum_{p=1}^{n-1} V_p = 1 + \sum_{p=1}^{n-1} \left(\prod_{j=1}^p (n-j) \right)$, while the number of operations of adding edges equals $\sum_{p=1}^{n-1} V_p = \sum_{p=1}^{n-1} \left(\prod_{j=1}^p (n-j) \right)$. Moreover, in each vertex, the LEXORDMAX algorithm is performed. Let E_{FAVOR} denote the number of elementary operations in FAVOR, thus inequality (1) is obtained. Similarly, we show that inequality (2) is satisfied. $\square$

Lemma A4. Let E_{ISIN} denote the number of operations performed by ISIN(v, S) algorithm. Then the following inequality holds

$$\bar{\bar{S}} \leq E_{\text{ISIN}} \leq \bar{\bar{S}}n.$$

Proof of Lemma A4. In the worst-case scenario the ISIN algorithm performs n comparisons for each vector from stack S . In the best-case scenario the ISIN algorithm performs one comparison for each vector from stack S . As a result, we obtain the inequality

$$\bar{\bar{S}} \leq E_{\text{ISIN}} \leq \bar{\bar{S}}n.$$

$\square$

Proof of Proposition 2. Let us observe that the number of elementary operations performed by NEUTRAL algorithm depends on the cardinality of sets NF and DNF . Let m denote the cardinality of set X , then we get $1 \leq \bar{\bar{NF}} \leq m$ and $1 \leq \bar{\bar{DNF}} \leq m$. The number of elementary operations performed by the ISIN algorithm employed in NEUTRAL is between $\bar{\bar{NF}} \cdot \bar{\bar{DNF}}$ and $\bar{\bar{NF}} \cdot \bar{\bar{DNF}} \cdot n$. Let $E_{\text{NEUTRAL,ISIN}}$ denote the number of elementary operations performed by ISIN in NEUTRAL and consider inequalities for $\bar{\bar{NF}}$ and $\bar{\bar{DNF}}$, then the pair of inequalities is obtained:

$$1 \leq E_{\text{NEUTRAL,ISIN}} \leq nm^2.$$

The numbers of operations performed by the algorithms FAVOR and DISFAVOR that are employed in the NEUTRAL algorithm satisfy inequalities (1) and (2). Let E_{NEUTRAL}

denote the number of elementary operations performed by NEUTRAL, then the following inequalities yield:

$$\begin{aligned}
 3 + 2 \sum_{p=1}^{n-2} \left(m + n + \frac{p}{\prod_{i=1}^{p-1} (n-i)} \right) 4n(n-1)! &\leq E_{NEUTRAL} \\
 &\leq 3 + 4 \sum_{p=1}^{n-1} \left(\prod_{j=1}^p (n-j) \right) \\
 &\quad + 2 \sum_{p=1}^{n-2} \left(2mn + \left(\frac{np^2}{2} + \frac{np}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n-i) \right) \\
 &\quad + 2(n-1)! \left(\frac{n^3}{3} + \frac{5n}{3} \right)
 \end{aligned}$$

This concludes the proof. $\square$

Corollary of Proposition 2. *If the size of the longest favoring sequential coalition is equal to n_f and the size of longest disfavoring sequential coalition is equal to n_d , $E_{NEUTRAL}$ will satisfy inequalities in the worst-case scenario*

$$\begin{aligned}
 3 + \sum_{p=1}^{n_f-2} \left(m + n_f + \frac{p}{\prod_{i=1}^{p-1} (n_f-i)} \right) + 2n_f(n_f-1)! & \\
 + \sum_{p=1}^{n_d-2} \left(m + n_d + \frac{p}{\prod_{i=1}^{p-1} (n_d-i)} \right) + 2n_d(n_d-1)! &\leq E_{NEUTRAL} \\
 &\leq 3 + 2 \sum_{p=1}^{n_f-1} \left(\prod_{j=1}^p (n_f-j) \right) \\
 &\quad + \sum_{p=1}^{n_f-2} \left(2mn_f + \left(\frac{n_f p^2}{2} + \frac{n_f p}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n_f-i) \right) \\
 &\quad + (n_f-1)! \left(\frac{n_f^3}{3} + \frac{5n_f}{3} \right) + 2 \sum_{p=1}^{n_d-1} \left(\prod_{j=1}^p (n_d-j) \right) \\
 &\quad + \sum_{p=1}^{n_d-2} \left(2mn_d + \left(\frac{n_d p^2}{2} + \frac{n_d p}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n_d-i) \right) \\
 &\quad + (n_d-1)! \left(\frac{n_d^3}{3} + \frac{5n_d}{3} \right).
 \end{aligned}$$

Proof of Corollary. Let us note that in the proof of Lemma 3 we assumed that the longest favoring (disfavoring) sequential coalition consisted of n coordinates. Therefore, if the size of the longest favoring sequential coalition were equal to n_f , the inequalities (1) in this Lemma would be in the form of

$$\begin{aligned}
 1 + \sum_{p=1}^{n_f-2} \left(m + n_f + \frac{p}{\prod_{i=1}^{p-1} (n_f-i)} \right) + 2n_f(n_f-1)! &\leq E_{FAVOR} \\
 &\leq 1 + 2 \sum_{p=1}^{n_f-1} \left(\prod_{j=1}^p (n_f-j) \right) \\
 &\quad + \sum_{p=1}^{n_f-2} \left(2mn_f + \left(\frac{n_f p^2}{2} + \frac{n_f p}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n_f-i) \right) \\
 &\quad + (n_f-1)! \left(\frac{n_f^3}{3} + \frac{5n_f}{3} \right).
 \end{aligned}$$

Moreover, if the size of the longest disfavoring sequential coalition were equal to n_d , the inequalities (2) in this Lemma would be in the form of

$$\begin{aligned} 1 + \sum_{p=1}^{n_d-2} \left(m + n_d + \frac{p}{\prod_{i=1}^{p-1} (n_d - i)} \right) + 2n_d(n_d - 1)! &\leq E_{FAVOR} \\ &\leq 1 + 2 \sum_{p=1}^{n_d-1} \left(\prod_{j=1}^p (n_d - j) \right) \\ &\quad + \sum_{p=1}^{n_d-2} \left(2mn_d + \left(\frac{n_d p^2}{2} + \frac{n_d p}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n_d - i) \right) \\ &\quad + (n_d - 1)! \left(\frac{n_d^3}{3} + \frac{5n_d}{3} \right) \end{aligned}$$

If we use these inequalities in the proof of Proposition 2, instead of inequalities (1) and (2) from Lemma 3, we will obtain

$$\begin{aligned} 3 + \sum_{p=1}^{n_f-2} \left(m + n_f + \frac{p}{\prod_{i=1}^{p-1} (n_f - i)} \right) + 2n_f(n_f - 1)! & \\ + \sum_{p=1}^{n_d-2} \left(m + n_d + \frac{p}{\prod_{i=1}^{p-1} (n_d - i)} \right) + 2n_d(n_d - 1)! &\leq E_{NEUTRAL} \\ &\leq 3 + 2 \sum_{p=1}^{n_f-1} \left(\prod_{j=1}^p (n_f - j) \right) \\ &\quad + \sum_{p=1}^{n_f-2} \left(2mn_f + \left(\frac{n_f p^2}{2} + \frac{n_f p}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n_f - i) \right) \\ &\quad + (n_f - 1)! \left(\frac{n_f^3}{3} + \frac{5n_f}{3} \right) + 2 \sum_{p=1}^{n_d-1} \left(\prod_{j=1}^p (n_d - j) \right) \\ &\quad + \sum_{p=1}^{n_d-2} \left(2mn_d + \left(\frac{n_d p^2}{2} + \frac{n_d p}{2} - \frac{p^3}{6} + \frac{p}{6} \right) \prod_{i=1}^{p-1} (n_d - i) \right) \\ &\quad + (n_d - 1)! \left(\frac{n_d^3}{3} + \frac{5n_d}{3} \right), \end{aligned}$$

which concludes the proof. $\square$

References

1. Łyko, J.; Rudek, R. A fast exact algorithm for the allocation of seats for the EU Parliament. *Expert Syst. Appl.* **2013**, *40*, 5284–5291. [\[CrossRef\]](#)
2. Arora, S.; Gupta, D.; Jain, S. An improvement in LaRSA and its implementation on allocation of seats to categories in an organization. *Indian J. Comput. Sci. Eng.* **2016**, *6*, 182–189.
3. Bateni, M.; Hajiaghay, M.; Immorlica, N.; Mahini, H. The cooperative game theory foundations of network bargaining games. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 67–78.
4. Goel, A.; Krishnaswamy, A.K. Implementing the Lexicographic Maxmin Bargaining Solution. *arXiv* **2018**, arXiv:1810.01042.
5. Ito, T.; Kakimura, N.; Kamiyama, N.; Kobayashi, Y.; Okamoto, Y. Efficient stabilization of cooperative matching games. *Theor. Comput. Sci.* **2017**, *677*, 69–82. [\[CrossRef\]](#)
6. Padmaja, D.L.; Vishnuvardhan, B. Comparative study of feature subset selection methods for dimensionality reduction on scientific data. In Proceedings of the 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, 27–28 February 2016; pp. 31–34.
7. Delgado-Márquez, B.; Kaeding, M.; Palomares, A. A more balanced composition of the European Parliament with degressive proportionality. *Eur. Union Politics* **2013**, *14*, 458–471. [\[CrossRef\]](#)
8. Dniestrzański, P. Alpha Proportionality and Penrose Square Root Law. *Eur. Proc. Soc. Behav. Sci.* **2016**, *10*. [\[CrossRef\]](#)
9. Grimmett, G. European apportionment via the Cambridge Compromise. *Math. Soc. Sci.* **2012**, *63*, 68–73. [\[CrossRef\]](#)
10. Martinez-Aroza, J.; Ramirez-Gonzalez, V. Several methods for degressively proportional allotments: A case study. *Math. Comput. Model.* **2008**, *48*, 1439–1445. [\[CrossRef\]](#)
11. Pukelsheim, F. 2010: Putting citizens first: Representation and power in the European Union. In *Institutional Design and Voting Power in the European Union*; Cichocki, M., Zyczkowski, K., Eds.; Ashgate Publishing: London, UK, 2010; pp. 235–253.

12. Ramirez-Gonzalez, V. Seat distribution in the European Parliament according to the Treaty of Lisbon. *Math. Soc. Sci.* **2012**, *63*, 130–135. [[CrossRef](#)]
13. Słomczyński, W.; Życzkowski, K. Mathematical aspects of degressive proportionality. *Math. Soc. Sci.* **2012**, *63*, 94–101. [[CrossRef](#)]
14. Pukelsheim, F. *Proportional Representation. Apportionment Methods and Their Applications*; Springer: Berlin/Heidelberg, Germany, 2014.
15. Cegiełka, K.; Łyko, J.; Rudek, R. Beyond the Cambridge Compromise algorithm towards degressively proportional allocations. *Oper. Res.* **2019**, *19*, 317–332. [[CrossRef](#)]
16. Dniestrzański, P. The proposal of allocation of seats in the European Parliament—The shifted root. *Procedia Soc. Behav. Sci.* **2014**, *124*, 536–543. [[CrossRef](#)]
17. Grimmett, G.; Laslier, J.-F.; Pukelsheim, F.; Ramirez-González, V.; Rose, R.; Słomczyński, W.; Zachariasen, M.; Życzkowski, K. The Allocation between the EU Member States of the Seats in the European Parliament. The Cambridge Compromise. European Parliament, Directorate-General for Internal Policies, Policy Department C: Citizen's Rights and Constitutional Affairs. PE 432.760. 2011. Available online: [https://www.europarl.europa.eu/RegData/etudes/note/join/2011/432760/IPOL-AFCO_NT\(2011\)432760_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/note/join/2011/432760/IPOL-AFCO_NT(2011)432760_EN.pdf) (accessed on 14 July 2021).
18. Habermas, J. Citizen and state equality in a supranational political community: Degressive proportionality and the pouvoir constituant mixte. *J. Common Mark. Stud.* **2017**, *55*, 171–182. [[CrossRef](#)]
19. Serafini, P. Allocation of the EU Parliament seats via integer linear programming and revised quotas. *Math. Soc. Sci.* **2012**, *63*, 107–113. [[CrossRef](#)]
20. Cegiełka, K.; Dniestrzański, P.; Łyko, J.; Maciuk, A.; Rudek, R. On ordering a set of degressively proportional apportionments. In *Transactions on Computational Collective Intelligence*; XXVII; Springer: Cham, Switzerland, 2017; pp. 53–62.
21. González, V.R.; Aroza, J.M.; García, A.M. Spline methods for degressive proportionality in the composition of the European Parliament. *Math. Soc. Sci.* **2012**, *63*, 114–120. [[CrossRef](#)]
22. Łyko, J.; Maciuk, A. Minimal and maximal representation of degressively proportional allocation. *Eur. Proc. Soc. Behav. Sci.* **2016**, *10*, 10–18.
23. Łyko, J.; Rudek, R. Operations research methods in political decisions: A case study on the European Parliament composition. *Comput. Math. Organ. Theory* **2017**, *23*, 572–586. [[CrossRef](#)]