Hamilton full favoring apportionments

Ion Bolun

Abstract

Aspects of full favoring of large beneficiaries in apportionments using Hamilton method are discussed. In this aim, the requirements of full favoring apportionments compliance with Hamilton method's solutions were defined. Subsequently, the $\mathrm{A_{HL}}$ algorithm for determining the Hamilton apportionments which fully favor large beneficiaries is described. Using this algorithm, calculations for two examples were performed. The obtained results confirm the opportunity of using the $\mathrm{A_{HL}}$ algorithm for the generation of Hamilton full favoring apportionments.

Keywords: algorithm, apportionment problem, disproportionality, Hamilton method, favoring of large beneficiaries.

1 Introduction

Often it is necessary to distribute a given number M of discrete entities of the same kind among n beneficiaries, in proportion to a numerical characteristic assigned to each of them V_i, $i = \overline{1,n}$. This is known as proportional apportionment (APP) problem [1, 2]. The integer character of this problem usually causes a certain disproportion of the apportionment $\{x_i,\ i = \overline{1,n}\}$ [1, 3], some beneficiaries being favored at the expense of the others. Such favoring leads to the increase of disproportionality of the apportionment. Therefore, reducing the favoring in question is one of the basic requirements when is choosing the APP method to be applied for apportionments.

In this aim, it is needed to estimate this property quantitatively. One approach is proposed in [5]. Another, the "total (full) favoring", is examined in [6]. In [6], it was shown that the frequency of full favoring in apportionments, for the widely used Hamilton [3], Sainte-Laguë [3], d'Hondt [3] and Huntington-Hill [4] methods, is strongly decreasing on n, becoming approx. 0 at $n \geq$ 7-10. Aspects of the guaranteed generation of Hamilton apportionments, which fully favor large beneficiaries (with higher V_i value) at larger values of n, are examined in this paper.

2 Essence of full favoring in apportionments

The notion of "total (full) favoring" of beneficiaries was introduced in [6] based on the definition of favoring of large or of small beneficiaries by an APP method given in [1] (see Definition (1)).

Definition 1. *In an apportionment, large beneficiaries are fully favored if*

$$\frac{x_i}{V_i} > \frac{x_j}{V_j}, \tag{1}$$

whenever $x_i > x_j$, where $(i, j) \in \{1, 2, 3, \ldots, n\}$ [6].

Usually, in one and the same apportionment some large and some small beneficiaries can be favored and, nevertheless, mainly large or, on the contrary, mainly small beneficiaries are favored. Therefore, in [5] it is proposed to use two different notions: "favoring" of large or of small beneficiaries and "full favoring" of large or of small beneficiaries, the second being a particular case of the first one. The compliance of an apportionment with requirements (1) is referred to "full favoring" of large beneficiaries. The larger notion of "favoring" is used when in an apportionment large beneficiaries are predominantly favored or, on the contrary, the small ones in sense of [5].

In order to identify whether apportionments that fully favor large beneficiaries can be obtained when applying the Hamilton APP method, it is necessary to know the compliance conditions of this method with requirements (1).

3 Compliance of Hamilton apportionments with requirements (1)

The required apportionments have to be Hamilton's ones and, at the same time, be compliant with requirements (1). The conditions for the compliance of an apportionment with the solution obtained by Hamilton method (Hamilton apportionment) are defined by Statement 1. First, let: $Q = V/M$; $V_i = a_iQ + \Delta V_i > 0$, $i = \overline{1,n}$; $\Delta M = (\Delta V_1 + \Delta V_2 + \Delta V_3 + \ldots + \Delta V_n)/Q$, $1 \leq l \leq n-1$ and $x_i > x_{i+1}$, $i = \overline{1,n-1}$. Of course, occur $0 \leq \Delta V_i < Q$, $i = \overline{1,n}$.

Statement 1. *The necessary conditions for the compliance of an apportionment $\{x_i, i = \overline{1,n}\}$, which fully favors large beneficiaries, with the solution obtained by Hamilton method are*

$$\Delta V_i > \Delta V_k, i = \overline{1,l}, k = \overline{l+1,n}. \tag{2}$$

Indeed, the Hamilton method apportionment rule states [3] that in addition to the already apportioned a_i entities, $i = \overline{1,n}$, the remained unapportioned $\Delta M = l$ entities should be apportioned by one to the first beneficiaries with the largest ΔV_j value. So, taking into account that $x_i > x_{i+1}, i = \overline{1,n-1}$, the relations $x_i = a_i + 1, i = \overline{1,l}$ and $x_i = a_i, i = \overline{l+1,n}$ should take place when favoring large beneficiaries, and that can be only if occurs (2). ⧫

It should be mentioned that Statement 1 establishes relationships between beneficiaries of two groups, $\{i = \overline{1,l}$ and $i = \overline{l+1,n}\}$, but not between beneficiaries within each of these groups if $n > 2$, needed to be established when analyzing the full favoring of large beneficiaries according to requirements (1).

It is well known that overall, on an infinity of apportionments, Hamilton method doesn't favor beneficiaries [1, 3]. But it can be particular Hamilton apportionments which fully favor large beneficiaries. The respective requirements are defined by Statement 2.

Statement 2. *If $n > 2$ and $l = \Delta M$, the conditions for the compliance of a Hamilton apportionment $\{x_i, i = \overline{1,n}\}$ with the requirement (1) of* *full favoring of large beneficiaries, in addition to the (2) ones, are*

$$\Delta V_i < \frac{a_i}{a_{i+1}} \Delta V_{i+1}, \tag{3}$$

where $i = \overline{l+1, n-1}$ *if* $a_n > 0$ *and* $i = \overline{l+1, n-2}$ *if* $a_n = 0$ *for* $1 = l < n-1$ *(Case L1),*

$$\Delta V_i < \frac{\Delta V_{i+1}(a_i+1) - Q(a_i - a_{i+1})}{a_{i+1}+1}, i = \overline{1, l-1} \tag{4}$$

for $1 < l = n-1$ *(Case L2) and both, (3) and (4), for* $1 < l < n-1$ *(Case L3).*

Indeed, one has $0 \leq \Delta V_i < Q, i = \overline{1, n}$. Let's begin with **Case L3**, divided into three subcases:

L3a) $x_i = a_i + 1, x_k = a_k + 1, i = \overline{1, l-1}, k = \overline{i+1, l}$;

L3b) $x_i = a_i + 1, x_k = a_k, i = \overline{1, l}, k = \overline{l+1, n}$;

L3c) $x_i = a_i, x_k = a_k, i = \overline{l+1, n-1}, k = \overline{i+1, n}$.

In **Subcase L3a**, according to (1) it should be $x_i/V_i > x_k/V_k$, that is $(a_i+1)/(a_iQ+\Delta V_i) > (a_k+1)/(a_kQ+\Delta V_k), i = \overline{1, l-1}, k = \overline{i+1, l}$, from where one has

$$\Delta V_i < \frac{\Delta V_k(a_i+1) - Q(a_i - a_k)}{a_k+1}, i = \overline{1, l-1}, k = \overline{i+1, l}. \tag{5}$$

Let's show that requirements (5) are transitive. From (5), for $k = i+1$ one has

$$\Delta V_i < \frac{\Delta V_{i+1}(a_i+1) - Q(a_i - a_{i+1})}{a_{i+1}+1}, i = \overline{1, l-1} \tag{6}$$

and, respectively,

$$\Delta V_{i+1} < \frac{\Delta V_{i+2}(a_{i+1}+1) - Q(a_{i+1} - a_{i+2})}{a_{i+2}+1}, i = \overline{1, l-2}. \tag{7}$$

Taking into account (7), requirement (6) can be transformed as follows

$$\Delta V_i < \frac{(a_i+1)\frac{\Delta V_{i+2}(a_{i+1}+1)-Q(a_{i+1}-a_{i+2})}{a_{i+2}+1} - Q(a_i - a_{i+1})}{a_{i+1}+1}$$
$$= \frac{\Delta V_{i+2}(a_i+1) - \frac{Q(a_{i+1}-a_{i+2})}{a_{i+1}+1}(a_i+1) - \frac{Q(a_i - a_{+1})}{a_{i+1}+1}(a_{i+2}+1)}{a_{i+2}+1}$$
$$= \frac{\Delta V_{i+2}(a_i+1) - Q(a_i - a_{i+2})}{a_{i+2}+1}, i = \overline{1, l-2}. \qquad (8)$$

So, if relations (6) and (7) take place, then relation (8) occurs, too. The same way, one can show that occurs

$$\Delta V_i < \frac{\Delta V_{i+j}(a_i+1) - Q(a_i - a_{i+j})}{a_{i+j}+1}, i = \overline{1, l-1}, j = \overline{1, l-i}. \qquad (9)$$

Thus, relations (5) are transitive and can be replaced by the (4) ones.▼

In **Subcase L3b**, according to (1), it should be $x_i/V_i > x_k/V_k$, that is $(a_i+1)/(a_iQ+\Delta V_i) > a_k/(a_kQ+\Delta V_k), i = \overline{1,l}, k = \overline{l+1,n}$, from where one has $\Delta V_k(a_i+1) > a_k(\Delta V_i - Q)$. Because of $0 \le \Delta V_i < Q$ and $\Delta V_k(a_i+1) \ge 0$, the requirements $\Delta V_k(a_i+1) > a_k(\Delta V_i - Q), i = \overline{1,l}, k = \overline{l+1,n}$ always take place, that's why Subcase L3b is not specified in Statement 2.▼

In **Subcase L3c**, according to (1), it should be $x_i/V_i > x_k/V_k$, that is $a_i/(a_iQ+\Delta V_i) > a_k/(a_kQ+\Delta V_k), i = \overline{l+1, n-1}, k = \overline{i+1, n}$, from where one has

$$\Delta V_i < \frac{a_i}{a_k}\Delta V_k, i = \overline{l+1, n-1}, k = \overline{i+1, n} \qquad (10)$$

if $a_n > 0$ and $\Delta V_n a_i > \Delta V_i a_n = 0$ if $a_n = 0, i = \overline{l+1, n-1}$. The last inequality always takes place, therefore it is not included in (3).

It is easy to show that requirements (10) are transitive. From (10), one has $\Delta V_i < \frac{a_i}{a_{i+1}}\Delta V_{i+1}$ and $\Delta V_{i+1} < \frac{a_{i+1}}{a_{i+2}}\Delta V_{i+2}$, from where $\Delta V_i < \frac{a_i}{a_{i+1}}\frac{a_{i+1}}{a_{i+2}}\Delta V_{i+2} = \frac{a_i}{a_{i+2}}\Delta V_{i+2}$. In the same way one can show that relations $\Delta V_i < \frac{a_i}{a_{i+j}} \Delta V_{i+j}, i = \overline{l+1, n-1}, j = \overline{1, n-i}$ occur. Thus, relations (10) are transitive and can be replaced by the (3) ones.▼

The proof for **Cases L1** and **L2**, taking into account proofs for Subcases L3a and L3c, are trivial. ♦

When generating apportionments that fully favor large beneficiaries, the inequalities

$$\Delta V_i > \frac{a_i}{a_{i-1}} \Delta V_{i-1}, i = \overline{l+2, n}, l = \overline{1, n-2}, \tag{11}$$

$$\Delta V_i > \frac{\Delta V_{i-1}(a_i + 1) + Q(a_{i-1} - a_i)}{a_{i-1} + 1}, i = \overline{2, l}, \tag{12}$$

equivalent to the (3) and (4) ones, are also useful.

4 Generating Hamilton apportionments

Based on Statements 1 and 2, the **$\mathbf{A_{HL}}$ algorithm** for the generation of Hamilton apportionments which fully favor large beneficiaries was elaborated. According to (11), the lower the value of ΔV_{l+1}, the lower the values of $\Delta V_i, i = \overline{l+2, n}$. Similarly, according to (12), the lower the value of ΔV_1, the lower the values of $\Delta V_i, i = \overline{2, l}$. Taking into account these observations, in Figure 1 the basic conceptual steps of the $\mathrm{A_{HL}}$ algorithm are shown, considering $V > M$ and that the value of ΔM is known.

At Steps 3 and 4 of the $\mathrm{A_{HL}}$ algorithm, to $\Delta V_i > 0, i = \overline{1, n}$ minimal possible values are allocated: at Step 3 – to $\Delta V_i, i = \overline{l+1, n}$ according to requirement (11) and beginning with the value of $\Delta V_{l+1} > 0$; at Step 4 – to $\Delta V_i, i = \overline{1, l}$ according to requirement (12) and beginning with the value of $\Delta V_1 > z = max\{\Delta V_{l+1}, \Delta V_{l+2}, \Delta V_{l+3}, \ldots, \Delta V_n\}$ because of requirement (2). If after these allocations one has $\Delta M > l$, that is $\Delta V > \Delta U$, where $\Delta U = \Delta M Q$, then the solution doesn't exist.

On the contrary, if $\Delta V < \Delta U$, then one has to increase ΔV aiming to reach $\Delta V = \Delta U$. Because of requirement (2), it is relevant to increase first, maximal possible, the values of $\Delta V_i, i = \overline{1, l}$ beginning with $\Delta V_l < Q$. This is done at Step 5 according to requirement (4).

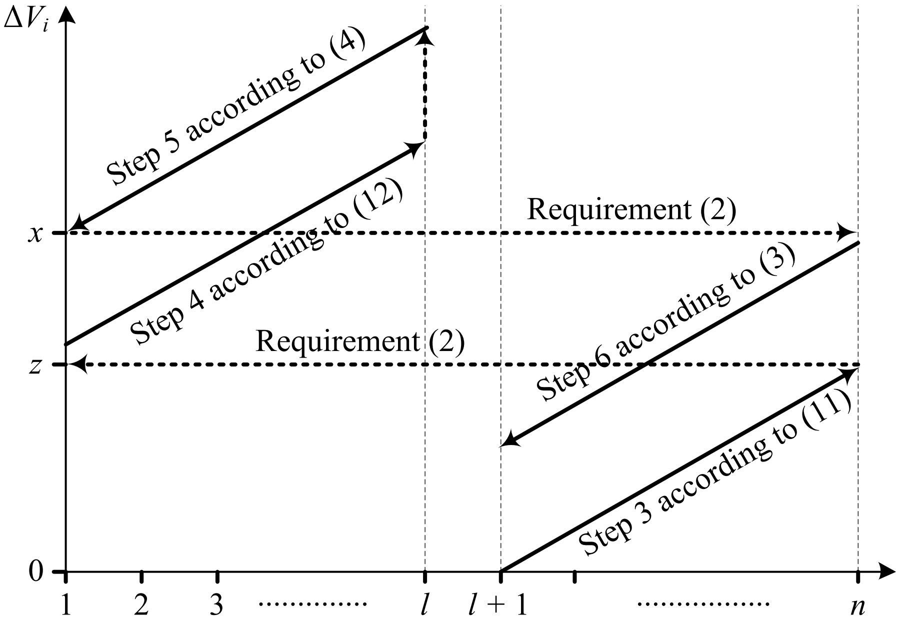

Figure 1. Basic steps of the $\mathrm{A_{HL}}$ algorithm.

But if the equality $\Delta V = \Delta U$ is not achieved at this step, then the last possibility to increase the ΔV value is the increase of $\Delta V_i, i = \overline{l+1, n}$ values beginning with $\Delta V_n < x = min\{\Delta V_i, i = \overline{1, l}\}$ because of requirement (2). This is done at Step 6 according to requirement (3).

It should be mentioned that in Figure 1 a continuous arrow doesn't reflect the relation between the values of ΔV_i and ΔV_{i-1}. It reflects the relation between ΔV_i and the respective function of:

1) ΔV_{i-1} (at Steps 3 and 4), that is $\Delta V_i > f_1(\Delta V_{i-1})$ according to requirement (11) and, respectively, the (12) one;

2) ΔV_{i+1} (at Steps 5 and 6), that is $\Delta V_i < f_2(\Delta V_{i+1})$ according to requirement (4) and, respectively, the (3) one.

The $\mathbf{A_{HL}}$ algorithm in details is described below.

1. Initial data are: $V, n, 1 \leq l \leq n-1, 1 \leq g \leq \lceil Q/n \rceil$ and $x_i > x_{i+1}, i = \overline{1, n-1}$.

2. $M := x_1+x_2+x_3+\ldots+x_n, Q := V/M, \Delta U := Ql; a_i := x_i-1, i = \overline{1, l}, a_i := x_i, i = \overline{l+1, n}$.

3. Based on (11), determining the preliminary, minimal possible, values of sizes $\Delta V_i \geq 0, i = \overline{l+1, n}$.
3.1. $i := l+1, \Delta V_i := \lfloor Qa_i \rfloor + 1 - Qa_i$. If $i = n$, then go to Step 4.
3.2. $i := i+1, \Delta V_i := \lfloor Qa_i + \Delta V_{i-1} a_i / a_{i-1} \rfloor + g - Qa_i$. If $\Delta V_i \geq Q$, then the solution doesn't exist. Stop.
3.3. If $i < n$, then go to Step 3.2.

4. Based on (12), determining the preliminary, minimal possible, values of sizes $\Delta V_i > 0, i = \overline{1, l}$.
4.1. $z := max\{\Delta V_{l+1}, \Delta V_{l+2}, \Delta V_{l+3}, \ldots, \Delta V_n\}; \Delta V := \Delta V_{l+1} + \Delta V_{l+2} + \Delta V_{l+3} + \ldots + \Delta V_n$.
4.2. $i := 1$. $\Delta V_i := \lfloor Qa_i + z \rfloor + g - Qa_i$. If $\Delta V_i \geq Q$, then the solution doesn't exist. Stop.
4.3. If $i = l$, then go to Step 5.
4.4. $i := i+1$. $\Delta V_i := \lfloor Qa_i + [\Delta V_{i-1}(a_i+1) + Q(a_{i-1} - a_i)]/(a_{i-1} + 1) \rfloor + g - Qa_i$. If $\Delta V_i \geq Q$, then the solution doesn't exist. Stop.
4.5. If $\Delta V_i \leq z$, then it is needed to minimally increase ΔV_i. $\Delta V_i := \lfloor Qa_i + z \rfloor + g - Qa_i$. If $\Delta V_i \geq Q$, then the solution doesn't exist. Stop.
4.6. If $i < l$, then go to Step 4.4.

5. Based on (4), ensuring $\Delta M = l$ by maximal possible increasing, if needed, the $\Delta V_i > 0, i = \overline{1, l}$ values.
5.1. $\Delta V := \Delta V + \Delta V_1 + \Delta V_2 + \Delta V_3 + \ldots + \Delta V_l$. If $\Delta V > \Delta U$, then the solution doesn't exist. Stop.
5.2. If $\Delta V = \Delta U$, then the solution is obtained. Go to Step 7.
5.3. $y := \Delta U - \Delta V, i := l$. If $Q - \Delta V_i > y$, then $\Delta V_i := \Delta V_i + y$ and the solution is obtained. Go to Step 7.
5.4. $h := \Delta V_i, \Delta V_i := \lceil Qa_l + Q \rceil - g - Qa_i, y := y - \Delta V_i + h$. If $l = 1$, then it is needed to increase the values of $\Delta V_i, i = \overline{l+1, n}$.

Go to Step 6.

5.5. $i := i - 1; h := \Delta V_i; \Delta V_i := \lceil Qa_i + [\Delta V_{i+1}(a_i + 1) - Q(a_i - a_{i+1})]/(a_{i+1} + 1)\rceil - g - Qa_i$. If $\Delta V_i < Q$, then:

5.5.1. If $\Delta V_i > h + y$, then $\Delta V_i := h + y$ and the solution is obtained. Go to Step 7.

5.5.2. $y := y - \Delta V_i + h$ and go to Step 5.8.

5.6. If $Q > h + y$, then $\Delta V_i := h + y$ and the solution is obtained. Go to Step 7.

5.7. $\Delta V_i := \lceil Qa_i + Q\rceil - g - Qa_i, y := y - \Delta V_i + h$.

5.8. If $i > 1$, go to Step 5.5.

6. Based on (3), ensuring $\Delta M = l$ by the maximal possible increase of the $\Delta V_i \geq 0, i = \overline{l + 1, n}$ values.

6.1. $x := min\{\Delta V_i, i = \overline{1, l}\}, i := n, h := \Delta V_i$. If $x > h + y$, then $\Delta V_i := h + y$ and the solution is obtained. Go to Step 7.

6.2. $\Delta V_i := \lceil Qa_i + x\rceil - g - Qa_i, y := y - \Delta V_i + h$.

6.3. If $i = l + 1$, then the solution doesn't exist. Stop.

6.4. $i := i - 1, h := \Delta V_i; \Delta Vi := min\{\lceil Qa_i + x\rceil; \lceil Qa_i + \Delta V_{i+1}a_i/a_{i+1}\rceil\} - g - Qa_i$. If $\Delta V_i > h + y$, then $\Delta V_i := h + y$ and the solution is obtained. Go to Step 7.

6.5. $y := y - \Delta V_i + h$. Go to Step 6.3.

7. Determining the $V_i, i = \overline{1, n}$ values. $V_i := Qa_i + \Delta V_i, i = \overline{1, n}$. Stop.

The obtained values of $V_i, i = \overline{1, n}$ can be checked by applying the Hamilton method. It should be noted that the affirmations "the solution doesn't exist" in the A$_{\text{HL}}$ algorithm are approximate, but very close to reality for $g = 1$. Parameter g is an integer, the value of which influences the minimal difference among the $x_i/V_i - x_{i+1}/V_{i+1}, i = \overline{1, n - 1}$ ones: the larger the value of g, the larger the mentioned difference. At the same time, the smaller the value of g, the higher the probability that the solution will be obtained.

Algorithm A$_{\text{HL}}$ was implemented in the computer application SIMAP. Examples 1 and 2 using SIMAP are described below.

Example 1 regarding the generation of a Hamilton apportionment which fully favors large beneficiaries. Initial data: $M = 279; n = 20; \Delta M = 10; V = 20000; g = 1$; the $x_i, i = \overline{1,n}$ values specified in Table 1.

Some results of calculus using SIMAP are systemized in Table 1.

Table 1. Calculations for the apportionment to Example 1

i	V_i	$10^{-7}x_i/V_i$	i	V_i	$10^{-7}x_i/V_i$
1	2145	139860	11	932	139485
2	1931	139824	12	789	139417
3	1788	139821	13	718	139276
4	1645	139818	14	575	139130
5	1574	139771	15	504	138889
6	1431	139762	16	433	138568
7	1360	139706	17	289	138408
8	1289	139643	18	217	138249
9	1146	139616	19	145	137931
10	1003	139581	20	86	116279

Example 2 regarding the generation of a Hamilton apportionment which fully favors large beneficiaries. Initial data are the same as in Example 1 with the only difference that $g = 3$. Some results of calculations using SIMAP are systemized in Table 2.

Data of Tables 1-2 were checked – the obtained apportionments are Hamilton ones. At the same time they comply with requirements (1). Thus, they fully favor large beneficiaries.

Comparing data in Tables 1 and 2, one can see that the obtained values of V_i and $x_i/V_i, i = \overline{1,n}$ differ. Using different values of g, one can obtain different solutions.

The minimal difference among the $x_i/V_i - x_{i+1}/V_{i+1}, i = \overline{1,n-1}$ ones is equal: to 3 if $g = 1$ and to 172 if $g = 3$. So, it is confirmed the fact that the larger the value of g, the larger the mentioned difference. Thus, if it is needed to increase this difference, one has to increase the

Table 2. Calculations for the apportionment to Example 2

i	V_i	$10^{-7}x_i/V_i$	i	V_i	$10^{-7}x_i/V_i$
1	2113	141978	11	932	139485
2	1904	141807	12	791	139065
3	1765	141643	13	722	138504
4	1626	141451	14	580	137931
5	1558	141207	15	512	136719
6	1419	140944	16	462	129870
7	1350	140741	17	318	125786
8	1281	140515	18	247	121458
9	1141	140228	19	175	114286
10	1001	139860	20	103	97087

value of g. But the value of g is limited from above by the value of $\lceil Q/n \rceil$ (approximately). In Examples 1 and 2, one has $Q = V/M = 20000/279 \approx 71.7$ and $\lceil Q/n \rceil = \lceil 71.7/20 \rceil = 4$. However, the attempt to obtain the solution using SIMAP for initial data of Examples 1 and 2 at $g = 4$ was unsuccessful.

5 Conclusions

In order to determine Hamilton apportionments which fully favor large beneficiaries, the A$_{\text{HL}}$ algorithm was elaborated. It guarantees the solution (if it exists), regardless of the value of n. Two examples of generating of such apportionments at $n = 20$ using the computer application SIMAP are described. In this context, it should be noted that in all 25 million variants of initial data with $n = 20$, for which the $V_i, i = \overline{1, n}$ values were generated stochastically at uniform distribution, none of the Hamilton apportionments fully favors the beneficiaries.

At the same time, it was identified that the results of calculations considerably depend not only on initial data $V, n, 1 \leq \Delta M \leq n-1$ and

$x_i, i = \overline{1,n}$, but also on parameter g value of the $\mathrm{A_{HL}}$ algorithm. The higher the g value ($1 \leq g \leq \lceil Q/n \rceil$), the larger the minimal difference among the $|x_i/V_i - x_{i+1}/V_{i+1}|, i = \overline{1, n-1}$ ones. But the maximal value of g, for which it is possible to obtain the solution, strongly depends on the value of ΔM, being small at small or large values of ΔM and large – at medium values of ΔM in the interval [1; n – 1].

References

[1] M.L. Balinski, H.P. Young. *Fair Representation: Meeting the Ideal of One Man, One Vote.* 2nd ed. Washington, DC: Brookings Institution Press, 2001.

[2] U. Kohler and J. Zeh. *Apportionment methods.* The Stata Journal, 2012, 12(3), pp. 375–392.

[3] M. Gallagher. *Proportionality, Disproportionality and Electoral Systems.* Electoral Studies, 1991, 10(1), pp. 33-51.

[4] E. Huntington. *A new method of apportionment of representatives.* Quart. Publ. Amer. Stat. Assoc., 1921, 17, pp. 859-1970.

[5] I. Bolun. *A criterion for estimating the favoring of beneficiaries in apportionments.* Proceedings of Workshop on Intelligent Information Systems WIIS2020, December 04-05, 2020. Chisinau: IMI, 2020, pp. 33-41.

[6] I. Bolun. *Total favoring in proportional apportionments.* Journal of Engineering Science, XXVIII, no. 1, 2021, pp. 47-60.

Ion Bolun

Technical University of Moldova
E–mail: `ion.bolun@isa.utm.md`