On the Computation of Semivalues for TU Games

Irinel Dragan

University of Texas at Arlington, Department of Mathematics, Arlington, Texas 76019-0408, USA

Abstract. In an earlier paper, [Dragan,2006a] , we proved that every Least Square Value is the Shapley Value of a game obtained by rescaling from the given game. In the paper where the Least Square Values were introduced by Ruiz, Valenciano and Zarzuelo, the authors have shown that the efficient normalization of a Semivalue is a Least Square Value. In the present paper, we develop the idea suggested by these two results, and we obtain a direct relationship between the Efficient normalization of a Semivalue and the Shapley Value [see also Dragan, 2006b]. The main tools for proofs were the so called Average per capita formulas we proved earlier for the Shapley Value [Dragan,1992] and for a Semivalue {Dragan and Martinez-Legas,2001]. Note that the connection between the Efficient normalization of a Semivalue and the Shapley Value has been used for computing a Semivalue, via a rescaling of the worth of coalitions in the given game. In this paper, the main purpose is to offer two other alternatives for the computation of a Semivalue, via the Shapley Value; beside the rescaling done before the computation, we consider rescalings within the computation. As seen above, this paper contains results from different sources, so that to make the paper self contained we shall be proving below our results together with the new results appearing here for the first time. Our proofs are algebraic, in opposition to those found in Ruiz et al., which are axiomatic. The direct connection between a Semivalue and the Shapley Value does not need any reference to the Least Square Values, which may well be unknown to the reader of the present paper.

In the first section, we prove the Average per capita formula for Semivalues, (Theorem 1), from which we derive our earlier Average per capita formula for the Shapley Value, to be used later. In the second section, we derive an Average per capita formula for the Efficient normalization of a Semivalue, by computing the efficiency term, (Theorem 2), as well as the main results, showing the connection between the Efficient normalization and the Shapley Value, (Theorems 3 and 4). In the third section, we discuss a first alternative for the computation of a Semivalue via the Shapley Value, illustrated in Example 1: we compute the needed ratios from the Average per capita formula for the Shapley Value and rescaling is done only \(n-1\) times, over these ratios. In the last section, we discuss a second alternative method for computing Shapley Values; the algorithm has been invented for computing Weighted Shapley Values and it will be adapted to the computation of the Semivalues. Some formulas are derived from this new method for computing the Weighted Shapley Values, based upon the null space of the Weighted Shapley Value operator, [Dragan,2008]. Note that the Semivalues are not Weighted Shapley Values. The Example 2 is illustrating the algorithm on the same 4-person game as before. The motivation for the present work was the fact that we know other works for computing the Shapley Value, but no computational work for Semivalues is known to us.

Key words: Shapley Value, Semivalue, Average per capita formula, Least Square Value, Weighted Shapley Value.
Now, we have \( s = n - 2 \), and we get the worth of coalitions of the game \( w \) obtained by rescaling, and the application of the algorithm, as
\[
\begin{align*}
    w(1) &= w(2) = w(3) = w(4) = 0, \\
    w(1, 2) &= w(1, 3) = w(1, 4) = w(2, 3) = w(2, 4) = w(3, 4) = 0, \\
    w(1, 2, 3) &= \frac{103}{24}, \\
    w(1, 2, 4) &= \frac{77}{24}, \\
    w(1, 3, 4) &= \frac{31}{24}, \\
    w(2, 3, 4) &= \frac{31}{24}, \\
    w(1, 2, 3, 4) &= 1.
\end{align*}
\]
We obtain the Efficient normalization of the Semivalue by computing the Shapley Value
\[
ESE(N, v) = \left( \frac{95}{144}, \frac{95}{144}, \frac{1}{48}, -\frac{49}{144} \right),
\]
like in Example 1; the Semivalue follows from the computation of \( \alpha = \frac{1}{48} \). Obviously, the Semivalue is the same.

Acknowledgements

The author is expressing his gratitude to Maria Hossu for related computer assistance, which contributed to the overall improvement of the paper.

References.


1. Average per Capita Formula for Semivalues

Let $G^N$ be the vector space of cooperative TU games with a fixed set of players $N$, $n=|N|$. Consider a nonnegative weight vector $p^n \in R^n$, satisfying the normalization condition

$$\sum_{s=1}^{n} \binom{n}{s-1} p^n_s = 1.$$  \hspace{1cm} (1)

The Semivalues associated with $p^n$ have been introduced axiomatically by P. Dubey, A. Neyman and R.J. Weber, (1981), as values on $G^N$ and even on more general structures, uniquely defined by a group of axioms suggested by the axioms of the Shapley Value. For $G^N$, they proved that a Semivalue associated with a weight vector $p^n$ is given by the formula

$$SE_i(N, \nu) = \sum_{s \in S \subseteq N} p^n_s [\nu(S) - \nu(S - \{i\})] \quad \forall i \in N,$$ \hspace{1cm} (2)

where $s = |S|$, and $p^n_s$ is the common weight of the marginal contributions for all coalitions of size $s$. We take this formula as the definition of Semivalues on $G^N$. To define the Semivalues on the union of all spaces $G^N$ when $N$ is arbitrary, we mean for player sets of different sizes, we need a sequence of nonnegative weight vectors $p^1, p^2, ..., p^n, ...$, all satisfying normalization conditions similar to (1), that is $p^1_1 = 1, p^2_1 + p^2_2 = 1, p^3_1 + 2p^3_2 + p^3_3 = 1, ...$ and so on. The definition of a Semivalue on $G^T$ is given by a formula similar to (2), where $N$ is replaced by $T$, $n$ by $t$, and $p^n$ by $p^t$. However, the sequence of weight vectors are supposed to satisfy what we call “inverse Pascal triangle” conditions

$$p^{t-s}_s = p^t_s + p^{t+1}_s, \quad s = 1, ..., t-1.$$ \hspace{1cm} (3)

It easy to see that if $p^n$ is given, and (3) hold, then all weight vectors $p^t, t \leq n$, are uniquely determined and they satisfy the corresponding normalization conditions (1).

Note the important fact that among the Semivalues, for $p^n_r = \frac{(s-1)!(n-s)!}{n!}$, we get the Shapley value, for $p^n_r = 2^{1-n}$, we get the Banzhaf value, and many other values are also Semivalues. Therefore, if we prove an Average per capita formula for a Semivalue, then we get such a formula for the Shapley Value, for the Banzhaf Value, and for other values, by choosing particular expressions of the weight vectors.

We call an Average per capita formula, any formula in which occur only the average worth of various coalitions, defined as follows:
\[ v_s = \frac{\sum_{|S| = s} v(S)}{\binom{n}{s}}, \quad v'_s = \frac{\sum_{|S| = s, i \in S} v(S)}{\binom{n-1}{s}}, \quad \forall i \in N, \quad s = 1, \ldots, n-1. \quad (4) \]

Clearly, \( v_s \) is the average worth of coalitions of size \( s \), while \( v'_s \) is the average worth of coalitions of size \( s \) which do not contain player \( i \). If we denote \( v_n = v(N) \), then there are \( n \) averages \( v_s \), and \( n(n-1) \) averages \( v'_s \), hence all together there are \( n^2 \) numbers associated with a given game. Let us introduce also new weights, defined for all \( t \leq n \) by
\[ q_s^t = \frac{p_s^t}{\gamma'_s}, \quad s = 1, \ldots, t, \quad (5) \]
where \( \gamma'_s = (t!)^{-1}(s-1)!(t-s)! \), that is the weights for the Shapley Value on \( G^t \).

**Theorem 1, [Dragan, and Martinez-Legas,2001]:** Let \( SE : G^n \to R^n \) be a Semivalue associated with a nonnegative weight vector \( p^n \) satisfying the normalization condition (1). Let \( q^n \) be the nonnegative weight vector defined by (5). Then, \( SE \) defined by (2), may be expressed in terms of the averages (4) and the weights (5) as
\[ SE_i(N, v) = q^n \frac{v_n}{n} + \sum_{s=1}^{n-1} q_s^n v_s - q_{s-1}^{n-1} v'_s, \quad \forall i \in N. \quad (6) \]

For \( q^n_s = 1, \ s = 1, \ldots, n \), that is \( p^n = \gamma^n \), we obtain:

**Corollary 1.** The Shapley Value of the game \( (N, v) \), is given by
\[ SH_i(N, v) = \frac{v_n}{n} + \sum_{s=1}^{n-1} \frac{v_s - v'_s}{s}, \quad \forall i \in N. \quad (7) \]

**Proof.**
For \( i \in N \) fixed, rewrite (2) as
\[ SE_i(N, v) = p^n_i v(N) + \sum_{S \in \mathcal{S}_C} p^n_S v(S) - \sum_{S \in \mathcal{S}_C} p^n_{S-\{i\}} v(S); \quad (8) \]
now, write the two sums separately as
\[ \sum_{S \in \mathcal{S}_C} p^n_S v(S) = \sum_{s=1}^{n-1} p^n_s \left( \sum_{|S| = s, i \in S} v(S) \right) = \sum_{s=1}^{n-1} p^n_s \left( \sum_{|S| = s} v(S) - \sum_{|S| = s, i \in S} v(S) \right), \quad (9) \]
and
\[ \sum_{S \in \mathcal{S}_C} p^n_{S-\{i\}} v(S) = \sum_{s=1}^{n-1} p^n_{s+1} \left( \sum_{|S| = s} v(S) \right). \quad (10) \]

From (8), (9) and (10), with notations (4), we obtain
\[ SE_i(N, v) = p_i^* v_i + \sum_{s=1}^{n-1} \left[ p_i^s \left( \begin{array}{c} n \\ s \end{array} \right) v_s - p_i^{s+1} \left( \begin{array}{c} n \\ s+1 \end{array} \right) v_{s+1} \right], \]

where we have used (3) for \( t = n \). If in (11) we introduce the new weights (5) by noticing that \( p_i^s \left( \begin{array}{c} n \\ s \end{array} \right) = s^{-1} q_i^s, \quad s = 1, \ldots, n \), we get (6).

Note that the new weights should satisfy the normalization condition \( \sum_{s=0}^{n} q_i^s = n \), derived from (1) and (5), and the inverse Pascal triangle conditions (3) become

\[ q_i^{s-1} = (1 - s^{-1}) q_i^s + s^{-1} q_i^{s+1}, \quad s = 1, \ldots, t - 1. \]

In the next section we derive a new Average per capita formula to express the normalizing term for the Efficient normalization of the Semivalue, and we shall derive also all needed results allowing the statement of the algorithm for its computation.

**2. Average per Capita Formula for the Efficiency Term**

In the paper where the Least Square Values have been introduced by Ruiz, Valenciano and Zarzuelo (1998), the authors defined what they called the Efficient normalization of a Semivalue \( SE \), associated with a nonnegative weight vector \( p^* = (p_i^*) \). This is the value \( ESE : G^N \rightarrow R^a \) written as

\[ ESE_i(N, v) = SE_i(N, v) + \alpha, \quad \forall i \in N, \]

with \( \alpha \) such that \( ESE \) is efficient, that is

\[ \alpha = \frac{1}{n} [v(N) - \sum_{i \in N} SE_i(N, v)]. \]

We call \( \alpha \) the efficiency term and we intend to derive an Average per capita formula for \( \alpha \), a fact which will be useful below theoretically and practically in the algorithm for computing the Semivalue. Of course, the normalization can be done in other ways, too. From (6), we obtain

\[ \sum_{j \in N} SE_j(N, v) = q_i^* v_i + \sum_{s=1}^{n-1} \frac{q_i^s v_s - q_i^{s+1} \sum_{j \in N} v_j}{s} = q_i^* v_i + n \sum_{s=1}^{n-1} \frac{(q_i^s - q_i^{s+1}) v_s}{s}, \]

where we have used the equality \( \sum_{j \in N} v_j = n v_s \), holding for all \( s = 1, \ldots, n - 1 \). In this way, from (13) and (14) we proved the result:

**Theorem 2.** The efficiency term for the additive normalization of a Semivalue is given by the Average per capita formula

\[ \alpha = (1 - q_i^*) \frac{v_i}{n} - \sum_{s=1}^{n-1} \frac{(q_i^s - q_i^{s+1}) v_s}{s}. \]

Notice that this formula is expressing \( \alpha \) in terms of the averages \( v_s \) only, so that it is easy to compute the efficiency term. Putting together the Average per capita formulas (6)
and (16) of the Theorems 1 and 2, we proved algebraically the main result for the Efficient normalization of a Semivalue:

**Theorem 3.** The Efficient normalization of a Semivalue associated with a nonnegative weight vector \( p^n = (p^n_s) \) is given by

\[
ESE_s(N, v) = \frac{v_s}{n} + \sum_{s=1}^{n-1} q_s^n \frac{v_s - v_i}{s}, \quad \forall i \in N,
\]

(17)

where \( q_s^n \) are expressed in terms of \( p^n \) and \( \gamma^n \) as

\[
q_s^n = \frac{p_s^n + p_{s+1}^n}{\gamma_s^n + \gamma_{s+1}^n}, \quad s = 1, ..., n-1.
\]

(18)

Note that (18) is derived from (5) for \( t = n - 1 \) and (3) for \( t = n \), taking into account that the Shapley weights satisfy also these conditions. Of course, for the Shapley Value we have \( \alpha = 0 \) and \( q_s^n = 1 \), so that we get (7). Instead, for the Banzhaf Value we get a new formula, where \( q_s^n = 2^{2-n} \left( \gamma_s^{n-1} \right)^{-1} \).

Note that Theorem 3 can be derived from the relationship axiomatically proved by Ruiz et al. between the Efficient normalization of a Semivalue and the Least Square Values and our relationship between the Least Square Values and the Shapley Values proved earlier.

In the present paper, it was no need of the Least Square Values, and therefore we have chosen the above proof.

Consider a game \( v \in G^N \) and rescale it by introducing the new game \( w \in G^N \):

\[
w(N) = v(N), \quad w(S) = q_s^{n-1} v(S), \quad \forall S \subseteq N.
\]

(19)

By formulas similar to (4), we have

\[
w_s = q_s^{n-1} v_s, \quad w_i = q_i^{n-1} v_i, \quad \forall i \in N, \quad s = 1, ..., n-1.
\]

(20)

Therefore, from (17) and (20) we obtain the right hand side in (7), for the new game \((N, w)\); we proved:

**Theorem 4.** The Efficient normalization of the Semivalue of a game \( v \in G^N \), associated with the weight vector \( p^n \in R^n_s \), is the Shapley Value of a new game \( w \in G^N \), obtained by a rescaling of the given game with factors \( q_s^{n-1} \) for the worth of coalitions of size \( s \), \( s = 1, ..., n-1 \), derived from the weight vector \( p^n \) and the Shapley weights by formulas (18).

Of course, if we subtract \( \alpha(S) = \alpha S, \forall S \subseteq N \), from the worth \( v(S) \), of the given game and use the linearity of the Shapley Value, we get also that the Semivalue is a Shapley Value. Notice that formula (17) of Theorem 3, as well as Theorem 4, are helpful in computing a Semivalue of a TU game via the Shapley Value, as it will be discussed in the next section.
3. The Average per Capita Formulas
and the Computation of Semivalues

As suggested by the formula (17), we can compute first the ratios needed in the
computation of the Shapley Value of the given game, and rescale the ratios; obviously,
we may rescale the given game first and further compute the Shapley Value as justified
by Theorem 4, as done in [Dragan,2006b, p.1547]. In both cases, we compute the
Efficient normalization of the Semivalue and from each component, we subtract the
number \( \alpha \). We give here an example based upon the above remark.

**Example 1.** Consider a four person simple game in which the winning coalitions are
\( \{1\}, \{2\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}, \{1,2,4\} \) and \( \{1,2,3,4\} \) and the weight vector
\( p^t \in \mathbb{R}^n \) is given by
\[
p^t = \left( \frac{1}{8}, \frac{1}{8}, \frac{1}{18}, \frac{1}{3} \right)
\]
From (3) for \( t = 4 \), we get the weight vector
\[
p^3 = \left( \frac{1}{4}, \frac{13}{72}, \frac{7}{18} \right); \text{ both } p^4 \text{ and } p^3 \text{ satisfy the corresponding normalization condition (1).}
\]
Taking into account that \( \gamma^3 = \left( \frac{1}{3}, -\frac{1}{6}, \frac{1}{3} \right) \), we use (5) for \( t = 3 \), to obtain
\[
q^3 = \left( \frac{3}{4}, \frac{13}{12}, -\frac{3}{6} \right),
\]
the vector which is weighing the ratios in (17), to get the Efficient normalization of the
Semivalues. Now, the usual computation of ratios in the Shapley Value Average per
capita formula for the given game provides
\[
\begin{align*}
\nu_1 &= \frac{1}{2}, & \nu^1_1 &= \frac{1}{3}, & \nu^3_1 &= \frac{2}{3}, & \nu^4_1 &= \frac{2}{3}, & \nu^4_1 - \nu^3_1 &= \left( \frac{1}{6}, \frac{1}{6}, -\frac{1}{6}, -\frac{1}{6} \right), \\
\nu_2 &= \frac{1}{2}, & \nu^1_2 &= \frac{1}{3}, & \nu^3_2 &= \frac{1}{3}, & \nu^4_2 &= 1, & \nu^2_2 - \nu^3_2 &= \left( \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, -\frac{1}{4} \right), \\
\nu_3 &= \frac{1}{2}, & \nu^1_3 &= 0, & \nu^3_3 &= \nu^4_3 &= 1, & \nu^3_3 - \nu^4_3 &= \left( \frac{1}{6}, \frac{1}{6}, -\frac{1}{6}, -\frac{1}{6} \right).
\end{align*}
\]
By using \( q^3 \) and formula (17), we obtain
\[
q^1_3 (\nu^1_1 - \nu^3_1) + q^2_3 (\nu^2_2 - \nu^3_2) + q^3_3 (\nu^3_3 - \nu^4_3) = \left( \frac{59}{144}, \frac{59}{144}, \frac{11}{48}, \frac{85}{144} \right),
\]
so that by adding \( \frac{1}{4} \) to each component, we have the Efficient normalization
\[
ESE(N, \nu) = \left( \frac{95}{144}, \frac{95}{144}, \frac{1}{48}, \frac{49}{144} \right).
\]
(23)
Now, we compute \( \alpha \) by means of (16); we need \( q^4 = \left( \frac{1}{2}, \frac{2}{3}, \frac{2}{3}, \frac{4}{3} \right) \) and \( q^3 \) computed
above, to use in (16) together with the averages \( \nu_1, \nu_2, \nu_3, \nu_4 \) to get \( \alpha = \frac{1}{48} \). We got
\[
SE(N, \nu) = ESE(N, \nu) - \alpha e = \left( \frac{23}{36}, \frac{23}{36}, 0, -\frac{13}{36} \right).
\]
(24)
Of course, we may verify the answer by means of formula (2).
As shown by the work in the Example 1, the algorithm has the operations:  

a) compute $q^{n-1}$, b) compute the ratios appearing in the Average per capita formula for the Shapley Value, of the given game, that is $\frac{v_s - v_s'}{s}$, for all $s = 1, ..., n-1$, followed by the weighted sum of these ratios; c) for all $i \in N$, add $\frac{v_i}{n}$, and we got the Efficient normalization of the Semivalue. d) Compute $\alpha$ and subtract it from each component, to get the Semivalue. Note that above we have used our Average per capita formula (7).

A different approach is offered by computing the Shapley Value via a recent algorithm, [Dragan, 2008], based upon the null space of the Shapley Value, computed for the first time in [Dragan et.al, 1989]. In fact the algorithm has been proved for the Weighted Shapley Value, which is not a Semivalue, and applied to the Shapley Value. To make the paper self contained let us describe instead the null space for the Shapley Value and the algorithm, applied to Semivalues, in the next section.

4. The Null Space of the Shapley Value and the Computation of Semivalues

In $G^N$, consider the set of games

$$W = \{W_s \in G^N : S \subseteq N, S \neq \emptyset\}, \forall S \subset N,$$

defined by

$$W_s(T) = \begin{cases} s & \text{if } T = S, \quad W_s(T) = -1, & \text{if } T = S \cup \{j\}, j \not\in S, \quad W_s(T) = 0, & \text{otherwise}. \end{cases}$$

For $S = N$ the middle case can not occur.

For example, when $N = \{1,2,3\}$, this set of games is

\[
\begin{array}{ccccccc}
W_1 & W_2 & W_3 & W_{12} & W_{13} & W_{23} & W_{123} \\
{\{1\}} & 1 & 0 & 0 & 0 & 0 & 0 \\
{\{2\}} & 0 & 1 & 0 & 0 & 0 & 0 \\
{\{3\}} & 0 & 0 & 1 & 0 & 0 & 0 \\
{\{1,2\}} & -1 & -1 & 0 & 2 & 0 & 0 \\
{\{1,3\}} & -1 & 0 & -1 & 0 & 2 & 0 \\
{\{2,3\}} & 0 & -1 & -1 & 0 & 2 & 0 \\
{\{1,2,3\}} & 0 & 0 & 0 & -1 & -1 & 3 \\
\end{array}
\]

It is obvious that this is a linearly independent vector set of 7 vectors in $G^{(1,2,3)} = R^7$, hence this is a basis for the space, and the same thing is true in general for $W$ in $G^N = R^{2^n-1}$. The similar basis of 15 vectors in $G^{(1,2,3,4)} = R^{15}$, will be used later in an example.

The computation of the Shapley Value for the games in basis W will be further needed and we shall do it now by means of the concept of Potential introduced by Hart and Mas Colell (1987). Recall that if the potential function is known, $P : G^N \rightarrow R$, then we have

$$SH_i(N, v) = P(N, v) - P(N - \{i\}, v), \forall i \in N.$$  \hspace{1cm} (27)

An useful result for our purpose is the explicit expression of the potential in term of the
coalitional form of the game offered also in the cited paper:

\[ P(N, \nu) = \sum_{S \subseteq N} \frac{(s-1)!(n-s)!}{n!} \nu(S). \]  

(28)

By (28), the potentials for the games (26) are

\[ P(N, W_s) = 0, \forall S \subseteq N, \quad P(N, W_N) = 1. \]  

(29)

We can prove the following

**Lemma.** The Shapley Value of the basic vectors in \( W \) are

\[ SH(N, W_s) = 0, \text{ if } |S| \leq n-2, \quad SH(N, W_{N-\{i\}}) = -e_i, \quad i = 1, \ldots, n, \]  

(30)

where \( e_i \) is a vector in \( R^n \) with the \( i \)-th component equal one, and the others equal zero, and

\[ SH(N, W_N) = e, \]  

(31)

where \( e \) has all components equal one.

**Proof.**

Now, by using the potentials (29) in (27) written for \( \nu = W_s \), and performing some elementary operations, we get that in the right hand side of (27) we have the first term equal one when \( S = N \), while the second term in the restricted game \( (N - \{i\}, \nu) \) equals zero, or the second term equal \(-1\) when \( S = N - \{i\} \), while the first term is zero; both are zero when \( S = N - \{j\}, j \neq i \). In all the other cases the Shapley Value equals zero. From the Lemma follows the following:

**Corollary 2.** The null space of the Shapley Value is generated by the basic games in the set

\[ \{W_s \in G^N : S \subseteq N, |S| \leq n-2\} \cup \{W_N + \sum_{i \in N} W_{N-\{i\}}\}. \]  

(32)

**Proof.**

This set has \( 2^n - n - 1 \) linearly independent games from the null space. As the range of the operator is \( R^n \), by a well known theorem of linear algebra, the entire null space is generated by the games in the set (32).

The algorithm that we intend to state is borrowing an idea from Maschler’s algorithm for computing a Shapley Value, namely the given game can be transformed into a new game in which the worth of the characteristic function are vanishing one at a time in each step. However, while Maschler’s algorithm is using sequential allocations of the worth of coalitions, until the entire worth is allocated, in our algorithm we are transporting the worth to coalitions of higher sizes, until the new game has zero worth for all coalitions of sizes smaller than or equal to \( n - 2 \). As it will be seen below, the Shapley Value can now be easily computed. Of course, we shall apply this to the rescaling of the given game, in order to compute the Efficient normalization of a Semivalue. The rescaling is easier because all worth we are rescaling belong to coalitions of the same size. Now, the transformation of the given game into a new game with the same Shapley Value is made by using combinations of the games from the null space of the operator,
which will not change the value. A similar approach has been used earlier for the computation of the Weighted Shapley Values and the Kalai-Samet Values (Dragan, 2008).

Let $s$ be an integer, $1 \leq s \leq n - 2$, such that either $s = 1$, or if $s \geq 2$, suppose that a game $v^{s-1} \in G^N$ derived from $v^0 = v$ is available, satisfying

$$v^{s-1}(T) = 0, \forall T \subset N, |T| \leq s - 1,$$

and

$$SH(N, v^{s-1}) = SH(N, v).$$

Suppose that $v^{s-1}(T) \neq 0$ for some coalition $T \subset N$ with $|T| = s$, and $s \leq n - 2$. Then, the derivation of the game $v^s$ satisfying conditions similar to (33) and (34), is explained by means of the result:

**Theorem 5.** Let $v^{s-1} \in G^N$ be a game satisfying (33) and (34) and $s \leq n - 2$. Then, the game

$$v^s = v^{s-1} - \sum_{T:|T|=s} \frac{v^{s-1}(T)}{|T|} W_T,$$

where $W_T$ are the games (26), satisfy the conditions obtained from (33) and (34) by changing $s$ into $s+1$.

**Proof.**

As the games $W_T$ with $|T| \leq n - 2$ belong to the null space of the Shapley Value, the equalities similar to (34) when $s$ is replaced by $s+1$, hold; it remains to show that the conditions similar to (33) still hold. The equality (35) can be written by components

$$v^s(U) = v^{s-1}(U) - \frac{1}{s} \sum_{T:|T|=s} v^{s-1}(T) W_T(U), \forall U \subseteq N.$$  \hspace{1cm} (36)

If $|U| \leq s-1$, then $W_T(U) = 0$ for all $U \subseteq N$, when $|T| = s$, hence from (33) and (36) we get $v^s(U) = 0$. If $|U| = s$, as $|T| = s$ in the sum, then we have $W_T(U) \neq 0$ only when $U = T$, and in this case we get $W_T(U) = W_T(T) = |T|$, so that from (36), taking into account (33), we have $v^s(U) = 0$. Hence, for all coalitions $U$ with $|U| \leq s$ we get $v^s(U) = 0$.

Note that in (35), or (36), taking into account the expressions (26) of basic games, we have in the sum non zero terms only for some coalitions $U$ with $|U| = s+1$; namely, this happens when $T = U \setminus \{j\}, j \in U$, and in this case $W_T(T \cup \{j\}) = -1$. Therefore, we obtain from Theorem 5, by means of (26), the formula which allows the computation of the characteristic function for the game obtained at the end of step $s$; as proved in Theorem 5, all worth for coalitions of sizes at most $s$ are zero, all worth for coalitions of sizes at least $s+2$ are unchanged, and the worth for coalitions of size $s+1$ are provided by the following:
**Corollary 3.** For any coalition $U$ of size $s+1$, the linear transformation (35), which makes the worth for coalitions of size $s$ equal to zero, gives

$$v'(U) = v^{s-1}(U) + \frac{1}{s} \sum_{j \in U} v^{s-1}(U - \{j\}).$$

(37)

The worth of coalitions of higher sizes remain the same.

Now, Theorem 4 and Corollary 3, will show how should we compute the Efficient normalization of a Semivalue: formula (37) should be applied to the game $(N,w)$ obtained after a rescaling, with the factor $q_{s-1}^{s}$, of the worth for coalitions of size $s$; as $|U| = s + 1$, formula (37) for $w \in G^N$, after dividing by $q_{s-1}^{s}$, becomes

$$v'(U) = v^{s-1}(U) + \frac{q_{s-1}^{s}}{s q_{s-1}^{s}} \sum_{j \in U} v^{s-1}(U - \{j\}), \forall U, |U| = s + 1.$$  

(38)

Hence, the step $s \leq n - 2$ is done as follows:

- the worth of coalitions of sizes 1 to $s - 1$, (if $s \geq 1$), and $s + 2$ to $n$, (if $s \leq n - 2$), are unchanged;
- the worth of coalitions of size $s$ become zero;
- the worth of coalitions of sizes $s + 1$ are computed by formula (37).

The algorithm ends when all coalitions of sizes at most $n - 2$ have worth zero, the worth of coalitions of size $n - 1$ have been computed in the last step, and the grand coalition has the initial worth. Now, after multiplying the worth of coalitions of size $n - 1$ by $q_{n-1}^{n-1}$, to get the corresponding worth of $w$, we can easily compute the Shapley Value, or even derive a special formula applicable in this case. Note that the factor multiplying the sum in (38) may be expressed in terms of the weights of the Semivalue and the Shapley coefficients. Of course, to compute the Semivalue from each component of the Efficient normalization we should subtract $\alpha$, given by formula (16).

**Example 2.** Let us return to the game considered in Example 1, and compute the Semivalue defined by the weight vector $\rho^s = (\frac{1}{8}, \frac{1}{8}, \frac{1}{18}, \frac{1}{3})$; the weight vector needed in computations is $q^s = (\frac{3}{4}, \frac{13}{12}, \frac{7}{6})$. In the first step, $s = 1$, the factor in front of the sum in formula (38) is $\frac{9}{13}$, and we can compute the worth of coalitions of size two

$$v^1(1,2) = \frac{31}{13}, \quad v^1(1,3) = \frac{22}{13}, \quad v^1(1,4) = \frac{9}{13}, \quad v^1(2,3) = \frac{22}{13}, \quad v^1(2,4) = \frac{9}{13}, \quad v^1(3,4) = 0.$$

In the second step, the factor in front of the sum in (38) is $\frac{13}{28}$, and we can compute the worth of coalitions of size three

$$v^2(1,2,3) = \frac{103}{28}, \quad v^2(1,2,4) = \frac{77}{28}, \quad v^2(1,3,4) = \frac{31}{28}, \quad v^2(2,3,4) = \frac{31}{28}. $$