Computing the Optimal Weights in a Utilitarian Model of Apportionment

Antonin Macé, Rafael Treibich

Computing the Optimal Weights in a Utilitarian Model of Apportionment

Antonin Macé*, Rafael Treibich†
École Polytechnique‡

October 2011

Abstract

We consider the model of optimal apportionment introduced by Koriyama et al. (2011 [8]) and suggest an iterative algorithm for the numerical optimization of the utilitarian social welfare function that is applied to the case of the European Union (EU). Our results confirm that the optimal system of weights exhibits a form of degressive proportionality.

1 Introduction

A committee representing countries or states of different sizes has to decide whether to accept or reject repeated proposals. Countries are given a fixed weight, and the final decisions are voted at the weighted majority : the proposal is accepted if the total weight of the countries which are in favor of the proposal exceeds a given threshold. The committee can either take the form of a council, such as the German Bundesrat, or a parliament, such as the European Parliament. In such a committee, how should the national weights (or number of seats) be distributed between the countries as a function of their respective populations?

The issue of fair apportionment arises in many intergovernmental organizations (EU, IMF,...) and federations (Germany, the United States). A natural way to proceed consists in allocating the weights in proportion to the countries' population sizes. However, despite its simplicity, the proportionality rule has often been criticized for being too harsh on smaller countries. Instead, politicians have argued in favor of the principle of degressive proportionality, requiring that the average weight per citizen be decreasing with the country's population size. Such a motivation appears clearly for example in the Treaty of Lisbon, which stipulates that "the representation of citizens shall be degressively proportional"[1]. We wonder if this political intuition is also supported by stronger normative arguments. Many of the models of fair apportionment do give support to the principle of degressive proportionality.

*antonin.mace[at]gmail.com
†rafael.treibich[at]gmail.com
‡We thank Jean-François Laslier, Yukio Koriyama, Tania Treibich, Margaux Luflade, Francis Bloch and two anonymous referees for helpful comments.
[1]See article 9.A of the Treaty of Lisbon [15].

Two main approaches to the issue have been taken in the literature. A first approach focuses on the concept of voting power and argues that the appropriate normative criterion for determining the national weights should be to equalize some measure of voting power across citizens. This path is followed, among others, by Penrose ([13] 1947), Felsenthal and Machover ([5] 1998) and Laruelle and Valenciano ([10] 2008) who all recommend some form of degressive proportionality. The second approach focuses on the expected utility individuals derive from a single decision of the committee and argues that the optimal weights should maximize a function of these utilities (a Rawlsian or a utilitarian social welfare function for example). Some of these models (Barbera and Jackson [1] (2006), Beisbart and Bovens 2007 [2]) give further support to the principle of degressive proportionality.
In a recent paper, Koriyama et al. ([8] 2011) suggest a new approach to the issue of fair apportionment. They argue that in such a context (a committee making repeated decisions), individuals do really care about the general probability that they will agree with the committee's final decision, their "probability of success". Following this psychological premise, Koriyama et al. introduce a concave utility function which takes the individual's probability of success as input and argue that the optimal set of weights should be determined as to maximize the sum of these individual utilities.

They show that the (real) optimal weights exhibit degressive proportionality.
In this paper, we suggest to tackle some computational issues that are not addressed in Koriyama et al.([8] 2011). Our final goal is to offer a numerical strategy to determine the optimal vector of weights, and to investigate to what extent the model actually supports the principle of degressive proportionality. The first issue concerns the probabilities of success, for which there is no closed form expression. We show that it is possible to compute the exact probabilities of success in a polynomial time, which then allows us to perform the numerical maximization without resorting to any approximation. The second issue concerns the optimization itself. The maximization set being too large, it is not possible to perform an exhaustive comparison of all the elements in the set. We suggest an iterative algorithm which always converges to a local optimum. Running this algorithm from random vectors of weights, we then select the highest utility local optimum as a plausible candidate for the global optimum. We find that the optimal vector of weights exhibits a form of degressive proportionality for a large class of concave utility functions. Furthermore, we identify a robust increasing relation between the concavity of the individual utility function and concavity of the curve supporting the optimal weights. Throughout the paper, we focus on the particular example of the European Parliament by fixing the total number of countries to 27 and total weight to 751[2].
The rest of the article is organized as follows. We start in Section 2, by presenting Koriyama et al.'s model of optimal apportionment. In Section 3, we explain how to compute the exact probabilities of success and suggest an algorithm to draw random vectors of weights with the uniform distribution, that will both serve for the numerical maximization. In Section 4, we describe the optimization algorithm and apply it to the case of the European Parliament.

2 A Model of Optimal Apportionment

We follow Koriyama et al.'s model of optimal apportionment (2011 [8]). There are c different countries $i \in C = \{1, ..., c\}$, each with population n_i. They are all represented in a committee which has to decide whether to accept $(D = 1)$ or reject $(D = 0)$ several proposals. Each country i is given a fixed

[2] As of June 2011, the European Parliament only contains 731 seats. However, the Treaty of Lisbon stipulates that this number will increase to 751 by 2014.

weight w_i which is used for all the decisions taken by the committee. For any decision, a country i can either be in favor or against the proposal. This choice is modeled as the realization of a Bernoulli variable X_i which takes the value 1 if the country is in favor and 0 if it is against. Each country has the same probability γ of being in favor of a proposal and the Bernoulli variables are assumed to be independent across countries. This independence assumption, although unrealistic, can be defended on the normative ground that we should not take into account the particular correlations between countries to apportion the weights. This argument is well explained in Laruelle and Valenciano ([9] 2005). Also, note that despite the independence assumption, there is still some degree of correlation between the countries' decisions when γ is different from 1/2. For any proposal, representatives vote for their country's preferred alternative, and the final decision is taken at the weighted majority: the proposal is accepted if the total weight of the countries in favor exceeds a given threshold. The objective of the paper is to suggest a rule allocating the weights between the countries.

A country is said to be successful whenever it is in favor of a proposal and the committee chooses to accept it or whenever it is against a proposal and it chooses to reject it. For a given vector of weights $w = (w_1, ..., w_n)$, country i's probability of success p_i thus writes :

$$\begin{aligned} p_i(w) &= \mathbb{P}(X_i = 1,\, D = 1) \;+\; \mathbb{P}(X_i = 0,\, D = 0) \\ &= \gamma \mathbb{P}(\sum_{j \neq i} w_j X_j \geq s - w_i) \;+\; (1-\gamma)\mathbb{P}(\sum_{j \neq i} w_j X_j < s) \end{aligned}$$

where the threshold s is equal to half of the total weight[3]: $s = (\sum_{i=1}^{c} w_i)/2$. The probability of success p_i corresponds to the expected frequency of decisions favorable to country i. When $\gamma = \frac{1}{2}$, the probabilities of success always lie in the interval $[\frac{1}{2}, 1]$ (countries with zero weight have a 1/2 probability of agreeing with the committee's decision).

We consider a simple version of Koriyama et al.'s model without intra-country heterogeneity: countries are assimilated to groups of interest in which all individuals share the same opinion[4]. Therefore, it is unnecessary to distinguish individuals within a given country, as they all enjoy the same probability of success. Also, it is assumed that individuals derive utility directly from the probability of success. This is a notable difference with other models of optimal apportionment (such as Laruelle and Valenciano (2010 [11]), Fleurbaey (2009 [6]) or Barbera and Jackson (2006, [1])) focusing on the expected utility individuals get from a single decision. Although the expected utility can in some cases be expressed as a function of the probability of success, the two approaches are not equivalent because the individual utility function is not linear. Applying a utilitarian social welfare function, Koriyama et al. conclude that the optimal set of weights should be chosen so as to maximize the following criterion :

$$U \;=\; \sum_{i=1}^{c} n_i \psi(p_i)$$

where $\psi(.)$ is an increasing concave utility function (assumed identical for all the individuals). The concavity of $\psi(.)$ can be interpreted either as the decrease in marginal satisfaction when the probability

[3] Koriyama et al. ([8]) show that it is always optimal in their model to fix the threshold to half of the total weight. Furthermore, it is the threshold that is currently used by the European parliament.

[4] In an other version of the model, Koriyama et al. relax this hypothesis by assuming there is a probability μ (greater than 1/2) that an individual agrees with his country's opinion. In this setting, for $\gamma = \frac{1}{2}$, the exact same computations also apply for a transformation of the utility function into an other increasing and concave utility function. As an example, for the utility functions that will be used in Section 4, the model with $\mu = 0.6$, $\psi = log(. - 0.4)$ and $\gamma = \frac{1}{2}$ is equivalent to the model with $\mu = 1$, $\psi = log(.)$ and $\gamma = \frac{1}{2}$. Consequently, we keep $\mu = 1$ for the rest of the article.

of success increases or as an aversion to ambiguity. It can also be related to the hypothesis of submodularity of inter temporal preferences (Koriyama et al. [8]). The concavity assumption is essential to the model: it is what differentiates it from the other utilitarian models of optimal apportionment and the reason why it is more intricate and difficult to solve[5].

The objective is to find the vector of weights w^* which maximizes the utilitarian welfare function U. Unfortunately, as there are no closed form expressions for the distribution of a weighted sum of Bernoulli variables, it is not possible to derive an analytical solution to the problem. Koriyama et al. show that the optimum vector of weights w^* is such that each country's weight w_i^* is proportional to the associated marginal utility $n_i\psi'(p_i(w^*)$, implying that the principle of degressive proportionality is satisfied at the optimum. However, they are not able to compute the actual solution, nor to evaluate its degree of degressive proportionality.

Our main objective in this paper is to perform the numerical optimization when integer constraints are taken into account. We apply our numerical strategy to the case of the European Union. In the following section, we explain how to compute the exact probabilities of success and generate random vectors of weights with the uniform distribution. These tools will be later useful for the numerical optimization.

3 A Numerical Toolbox

3.1 Computational Structure of the Model

We restrict our analysis to integer weights. We denote by $\Gamma(c, \bar{W})$ the set of vector of weights of size c (the number of countries) which sum up to $\bar{W}$ (the total weight) :

$$\Gamma(c, \bar{W}) = \{w \in \mathbb{N}^c \mid \sum_{i=1}^{c} w_i = \bar{W}\}$$

In combinatorics, this set is usually referred to as the set of compositions of W with c positive parts. When $c = 27$ and $\bar{W} = 751$ (as for the European Parliament), its cardinal is around 10^{48}. In fact, as we show below, the solution is necessarily a non decreasing vector of weights, so we will only have to consider the subset of non decreasing vectors of weights. We give the corresponding definition :

Definition 1. *We say that a vector of weights is non decreasing if for any i and j in C :* $[\ n_i < n_j \quad \Rightarrow \quad w_i^* \le w_j^*\ ]$

We denote $\Gamma^+(c, \bar{W})$ the set of non decreasing vectors of weights of size c which sum up to $\bar{W}$.

$$\Gamma^+(c, \bar{W}) = \{w \in \Gamma(c, \bar{W}) \mid w \text{ non decreasing}\}$$

In combinatorics, this set is usually referred to as the set of partitions of $\bar{W}$ with c positive parts. We show that any solution w^* to the optimal apportionment problem is necessarily a non decreasing vector of weights.

Proposition 1. $\underset{w \in \Gamma(c,\bar{W})}{\arg\max}\ U(w) \ \subset\ \Gamma^+(c, \bar{W})$

[5] When ψ is linear, it can be shown that the solution of the maximization problem is the vector of proportional weights (Koriyama et al. [8]).

Proof. Let w^* be a solution to the optimal apportionment problem, and let i and j be two countries such that $n_i < n_j$. Assume $w_i^* > w_j^*$, and consider w' defined by $w_k' = w_k^*$ for any $k \neq i, j$, $w_i' = w_j^*$ and $w_j' = w_i^*$.
Then, for any $k \neq i, j$:

$$p_k(w') = \gamma \mathbb{P}(\sum_{l \neq k} w_l' X_l \geq s - w_k') \ + \ (1-\gamma)\mathbb{P}(\sum_{l \neq k} w_l' X_l < s) = p_k(w^*)$$

While for i and j:

$$p_i(w') = p_j(w^*) \ \text{ and } \ p_j(w') = p_i(w^*)$$

So that $U(w') = \sum_{i \in C} n_i \psi(p_i(w')) > \sum_{i \in C} n_i \psi(p_i(w^*)) = U(w^*)$ (since the utility function $\psi(.)$ is assumed increasing). Therefore, we get a contradiction with w^* being a solution to the optimal apportionment problem and conclude that $w_i^* \leq w_j^*$. $\square$

Therefore, we can restrict our attention to the function $U : \Gamma^+(C, \bar{W}) \to \mathbb{R}$. At the optimum, bigger countries will necessarily have larger (or equal) weights than smaller ones. Unfortunately, the set $\Gamma^+(C, \bar{W})$ is still too large (when $c = 27$ and $\bar{W} = 751$, its cardinal is around 10^{22}), to allow for an exhaustive comparison of all its elements. We will see in section 4 what kind of strategy can be used to perform the numerical optimization in a non exhaustive way.
In order to compute the social utility associated with any given vector of weights w, we first need to compute the corresponding vector of probabilities $p(w) = (p_1(w), ..., p_c(w))$. These probabilities depend on the particular choice for the probabilistic model. Unfortunately, since there is no closed form expression in our model for the probabilities of success, their computation is not straightforward. We will show in the next subsection how these probabilities can be computed numerically using the method of generating functions. Once we have the vector of probabilities, the computation of the social utility $U(w)$ only depends on the choice of the individual utility function ψ and on the vector of population $n = (n_1, ..., n_c)$. The computational structure of the model is summarized in the figure below.

Figure 1. Computational Structure of the Model

$$\begin{array}{c}\text{Vector of Weights}\\ w=(w_k)_{k=1\dots c}\end{array} \xrightarrow[\substack{\textit{Probabilistic}\\ \textit{Model}}]{\text{Step 1}} \begin{array}{c}\text{Probabilities of}\\ \text{Success}\\ p(w)=(p_k(w))_{k=1\dots c}\end{array} \xrightarrow[\substack{\textit{Population } \{n_i\}_{i\in C},\\ \textit{Utility function } \psi(.)}]{\text{Step 2}} U(w)$$

The maximization set $\Gamma^+(c, \bar{W})$ being too large for an exhaustive comparison of all its elements, we have to perform stochastic simulations and statistical analysis. To that purpose, we need to generate random vectors in $\Gamma^+(c, \bar{W})$ with the uniform distribution. In the following two subsections, we explain how to compute the exact probabilities of success and how to generate random vectors of weights with the uniform distribution.

3.2 Computing the Exact Probabilities of Success

Although the exact probabilities of success $\{p_i\}_{i \in C}$ cannot be given a closed form expression, it is possible to compute their values numerically. Let $w = (w_1, ..., w_c)$ be any vector of weights in $\Gamma(c, \bar{W})$. In order to determine the probability of success p_i we need to compute $\mathbb{P}(\sum_{j \neq i} w_j X_j \geq s - w_i)$ and $\mathbb{P}(\sum_{j \neq i} w_j X_j < s)$. For any possible realization of $X_{-i} = (X_j)_{j \neq i}$, we define S_{-i} as the subset of individuals in $C\backslash\{i\}$ which are in favor of the proposal: $S_{-i} = \{j \in C\backslash\{i\} | X_j = 1\}$. Any such coalition S_{-i} has a probability $\gamma^{|S_{-i}|}(1-\gamma)^{c-1-|S_{-i}|}$ of occurring. Therefore, we can write:

$$\begin{aligned}\mathbb{P}(\sum_{j\neq i} w_j X_j \geq s - w_i) &= \sum_{S \in \mathcal{P}(C\backslash\{i\})} \gamma^{|S|}(1-\gamma)^{c-1-|S|} \mathbf{1}_{\sum_{j\in S} w_j \geq s - w_i} \\ &= \sum_{l=0}^{c-1} \gamma^l (1-\gamma)^{c-1-l} \left| \left\{ S \in \mathcal{P}_l(C\backslash\{i\}) \mid \sum_{j\in S} w_j \ \geq \ s - w_i \right\} \right|\end{aligned}$$

where $\mathcal{P}_l(C\backslash\{i\})$ denotes the set of all subsets of $C\backslash\{i\}$ of size l. We conclude that:

$$\begin{aligned}p_i \ = \ \gamma \sum_{l=0}^{c-1} \gamma^l (1-\gamma)^{c-1-l} & \left| \left\{ S \in \mathcal{P}_l(C\backslash\{i\}) \mid \sum_{j\in S} w_j \ \geq \ s - w_i \right\} \right| \\ + (1-\gamma) \sum_{l=0}^{c-1} \gamma^l (1-\gamma)^{c-1-l} & \left| \left\{ S \in \mathcal{P}_l(C\backslash\{i\}) \mid \sum_{j\in S} w_j \ < \ s \right\} \right|\end{aligned}$$

In order to compute these probabilities, we need to count for each possible coalition size l, the number of coalitions in $\mathcal{P}_l(C\backslash\{i\})$ for which the sum of weights is either larger than $s - w_i$ or less than s. A direct computation would require an extremely high number of operations ($O(c2^{c-1})$) and would therefore take a very long time to proceed. In order to avoid this difficulty, we follow the method of generating functions which allows to compute the probabilities of success in a much faster way (polynomial time). This method is well known in enumerative combinatorics and was first applied to cooperative game theory in 1962 by Mann and Shapley (Leech 2002 [12]).
Let's consider the polynomial P given by :

$$P(x,y) \ = \ \prod_{j\neq i}(1 + x^{w_j} y) \ = \ \sum_{m=0}^{\bar{W}^{-i}} \sum_{l=0}^{c-1} d_{m,l}\, x^m y^l$$

where $\bar{W}^{-i} = \sum_{j\neq i} w_j$. The coefficients $d_{m,l}$ give the number of coalitions in $\mathcal{P}(C\backslash\{i\})$ of size l for which the sum of weights equals m. The probability of success now writes :

$$p_i \ = \ \gamma \sum_{l=0}^{c-1} \gamma^l (1-\gamma)^{c-1-l} \left[\sum_{m=s-w_i}^{\bar{W}^{-i}} d_{m,l} \right] \ + \ (1-\gamma) \sum_{l=0}^{c-1} \gamma^l (1-\gamma)^{c-1-l} \left[\sum_{m=0}^{s-1} d_{m,l} \right].$$

We denote by w^{-i} the vector of size $(c-1)$ obtained by discarding w_i from w. The coefficients $d_{m,l}$ can be computed iteratively by considering the following sequence of polynomials :

$$P^k(x,y) = \prod_{j=1}^{k} (1 + x^{w_j^{-i}} y) \qquad \forall\, k \in \{0, ..., c\}$$

Noting that $P^k(x,y) = P^{k-1}(x,y)(1 + x^{w_k^{-i}} y)$, we get the following iterative relation between the polynomial coefficients of P^k and P^{k-1} :

$$d^k_{m,l} = d^{k-1}_{m,l} + d^{k-1}_{m-w_k^{-i},l-1} \qquad \forall k \in \{1, ..., c\}$$

Using the fact that $d^0_{0,0} = 1$, $d^0_{m,l} = 0$ for all $(m, l) \neq (0, 0)$ and $d_{m,l} = d^c_{m,l}$, we can compute the desired probabilities $\{p_i\}_{i\in C}$. The corresponding matlab program is given in the Appendix (A1). For a given vector of weights, the process of computing the corresponding probabilities of success now only requires $O(c^3 \sum_{i=1}^{c} \bar{W}_{-i}) < O(c^4\bar{W})$ operations. When γ is equal to 1/2 (which will be our assumption in what follows), the process becomes even faster because we do not have to distinguish the coalitions by sizes anymore (since $(1/2)^l(1 - 1/2)^{c-l} = (1/2)^c$). The required number of operations to compute the probabilities of success decreases to $O(c^3\bar{W})$.
It is important to note that this algorithm only works for vectors of integer weights[6]. In fact, it is still possible to estimate the probabilities of success for non integer weights, by generating random profiles of votes $\{X_i\}_{i...c}$ for a large number of decisions and computing the empirical frequency of success for each country. Unfortunately, this process appears to converge slowly. For example, in the case of the European Parliament, it takes around 10^5 iterations to obtain the exact probabilities of success with 2 significant digits. Therefore, it would be impossible to use it for the maximization of the social welfare.

3.3 Generating Random Vectors of Weights

A simple and well known algorithm to generate random partitions of integers writes as follows:

Algorithm Alg0

Step 1. Draw (c-1) random variables $(X_1, ..., X_{c-1})$ independently with the uniform distribution over the set $\{0, .., \bar{W}\}$.

Step 2. Form the vector Y by rearranging X's elements in increasing order.

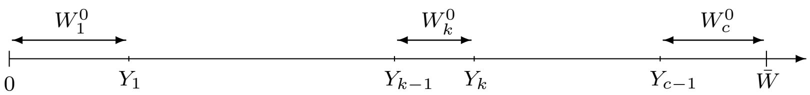
Step 3. Define the vector W^0 by setting: $W^0_1 = Y_1$, $W^0_k = Y_k - Y_{k-1}$ for all $k \in \{2, ..., c-1\}$ and $W^0_c = \bar{W} - Y_{c-1}$.

Step 4. Form the vector W by rearranging W^0's elements in increasing order.

The figure below represents the structure of algorithm Alg0.

[6] The algorithm can be adapted to compute the probabilities of success for vectors of rational weights. Assume there exists $\{(a_i, b_i)\}_{i\in C}$ such that $w_i = \frac{a_i}{b_i}$, $\forall i \in C$ and denote b the least common multiple of $\{b_i\}_{i\in C}$. Then, define the vector w' by setting $w'_i = bw_i$, $\forall i \in C$ and compute the corresponding probabilities of success $\{p_i(w')\}_{i\in C} = \{p_i(w)\}_{i\in C}$ using the method of generating functions. Note that if b is very large, the process could take a lot of time.

Figure 2. *Algorithm Alg0*



Unfortunately, this algorithm does not generate all vectors in $\Gamma^+(c, \bar{W})$ with the same probability. The basic reason is that, for a given vector of weights which is non decreasing but not increasing, there exists fewer non ordered vectors of weights that generate it once they are reordered. For example, while there exists $c!$ such different vectors for a strictly increasing vector there only exists one for the constant vector (himself). We denote by $\Gamma_*^{++}(c, \bar{W})$ the subset of increasing vectors of weights for which the minimum weight is strictly larger than zero :

$$\Gamma_*^{++}(c, \bar{W}) = \{w \in \Gamma^+(c, \bar{W}) \mid w \text{ increasing and } \forall k,\ w_k > 0\}.$$

We get the following result :

Proposition 2. *Algorithm Alg0 generates :*

- *all vectors in $\Gamma_*^{++}(c, \bar{W})$ with the same probability.*
- *all vectors in $\Gamma^+(c, \bar{W}) \backslash \Gamma_*^{++}(c, \bar{W})$ with a smaller probability than those in $\Gamma_*^{++}(c, \bar{W})$.*

Proof. Let w a vector in $\Gamma^{++}(c, \bar{W})$. We compute the probability that the vector w will be generated at step 4 of Algorithm Alg0 :

$$\begin{aligned}
\mathbb{P}(W_1 = w_1, ..., W_c = w_c) &= c!\, \mathbb{P}(W_1^0 = w_1, ..., W_c^0 = w_c) \\
&= c!\, \mathbb{P}(Y_1 = w_1, ..., Y_{c-1} = \sum_{k=1}^{c-1} w_k, Y_{c-1} = \bar{W} - w_c) \\
&= c!\, \mathbb{P}(Y_1 = w_1, ..., Y_{c-1} = \sum_{k=1}^{c-1} w_k) \\
&= c!(c-1)!\, \mathbb{P}(X_1 = w_1, ..., X_{c-1} = \sum_{k=1}^{c-1} w_k) \\
&= \frac{c!(c-1)!}{(\bar{W}+1)^{c-1}}
\end{aligned}$$

This probability is the same for all the vectors in $\Gamma^{++}(c, \bar{W})$. The first equality comes from the fact that there are exactly $c!$ different vectors W^0 that can generate a given vector W when reordered in increasing order and they are all equiprobable. This property is due to the fact that the vector w is increasing. The second equality follows from the definition of W^0. The third equality comes from the fact that $\sum_{k=1}^{c-1} w_k = \bar{W} - w_c$, since $w \in \Gamma(c, \bar{W})$. The fourth equality comes from the fact that there are exactly (c-1)! different vectors X generating a given vector Y when reordered in increasing order and they are all equiprobable. Indeed, the vector w is in $\Gamma^{++}(c, \bar{W})$, so for all k, $w_k > 0$, implying that the vector $(\sum_{l=1}^{k} w_l)_{k=1...(c-1)}$ is increasing. Finally, the last equality is given by the definition of the vector X.

We conclude Algorithm Alg0 generates vectors with equal probability on the set $\Gamma^{++}(c, \bar{W})$. If a vector w is in $\Gamma^+(c, \bar{W}) \backslash \Gamma_*^{++}(c, \bar{W})$, then:

- Either w is not (strictly) increasing, and the first equality above becomes a strict inequality ($<$).

- Or (and) there exists some k such that $w_k = 0$, and the vector $(\sum_{l=1}^{k} w_l)_{k=1...(c-1)}$ is not (strictly) increasing, so the fourth equality above becomes a strict inequality ($<$).

This achieves the proof.

$\square$

We understand if we slightly modify Algorithm Alg0, by rejecting all vectors not being in $\Gamma_*^{++}(c, \bar{W})$, we can generate vectors uniformly on this set. In order to generate vectors in $\Gamma^+(c, \bar{W})$ with the uniform distribution, we first observe that there exists a bijective mapping between $\Gamma^+(c, \bar{W})$ and $\Gamma_*^{++}(c, \bar{W} + \frac{c(c+1)}{2})$:

Lemma 1. *The following mapping is a bijection :*

$$\left| \begin{array}{ccc} \Gamma^+(c, \bar{W}) & \to & \Gamma_*^{++}(c, \bar{W} + \frac{c(c+1)}{2}) \\ (w_k)_{k=1...c} & \mapsto & (w_k + k)_{k=1...c} \end{array} \right.$$

Proof. Straightforward. $\square$

Using the inverse of this bijection, we can generate vectors in $\Gamma^+(c, \bar{W})$ with the uniform distribution. The modified algorithm writes as follows.

Algorithm Alg1

Step 1. Draw $(c-1)$ random variables $(X_1, ..., X_{c-1})$ independently with the uniform distribution over the set $\{0, .., \bar{W} + \frac{c(c+1)}{2}\}$

Step 2. If the realization of vector X is such that $X_i \neq X_j$ for all i and j, $i \neq j$, form the unique vector Y obtained by rearranging its elements in increasing order. Otherwise, go back to step 1.

Step 3. Define the vector W^0 by setting: $W_1^0 = Y_1$, $W_k^0 = Y_k - Y_{k-1}$ for all $k \in \{2, ..., c-1\}$ and $W_c^0 = \bar{W} + \frac{c(c+1)}{2} - Y_{c-1}$.

Step 4. If the vector W^0 is such that $W_i^0 \neq W_j^0$ for all i and j, $i \neq j$, form the unique vector W^1 obtained by rearranging its elements in increasing order. Otherwise, go back to step 1.

Step 5. Form the vector W by setting:

$$W_k = W_k^1 - k \quad \forall\, k \in \{1, ..., c\}$$

Proposition 3. *Algorithm Alg1 generates vectors uniformly over the set* $\Gamma^+(c, \bar{W})$.

Proof. We know from proposition 2 that any vector $W^0 \in \Gamma_*^{++}(c, \bar{W} + \frac{c(c+1)}{2})$ has an equal probability of being generated at the end of step 4. Using lemma 1, we conclude any vector $W \in \Gamma^+(c, \bar{W})$ has an equal probability of being generated at the end of step 5. $\square$

The corresponding matlab program is given in the Appendix (A2).

4 Optimization

4.1 The Algorithm

The maximization set $\Gamma^+(c, \bar{W})$ being too large, it is impossible to compare the utility of all its elements. We suggest an iterative algorithm that always converges to a local optimum. Starting from an initial vector of weights, the algorithm looks for random neighboring improvements until it reaches a vector for which there exists no such improvements (a local optimum). Running the algorithm from various random vectors of weights, we then select the highest utility local optimum as a plausible candidate for the global optimum. First, we give a few definitions.

Definition 2. *Let w and w' be two vectors of weights in $\Gamma^+(c, \bar{W})$. We say that w and w' are neighbors if there exists i and j in C such that: $w'_i = w_i + 1$, $w'_j = w_j - 1$ and $w'_k = w_k$ for all $k \neq i, j$.*

We denote $C(w)$ the set of w's neighboring vectors. Its cardinal is less than or equal to $c(c-1)$. We will say that a vector is locally optimal if its utility is larger than or equal to the utility of all his neighbors.

Definition 3. *Let w be a vector of weights in $\Gamma^+(c, \bar{W})$. We say that w is locally optimal if for any w' in $C(w)$, $U(w) \geq U(w')$.*

We suggest the following maximization algorithm :

Algorithm Alg2

Step 1. Use algorithm (Alg1) to draw a random vector w^0 in $\Gamma^+(c, \bar{W})$.

Step 2. Draw a random vector w from the set of w^0's neighbors, $C(w^0)$. If w's utility is larger than w^0's utility, replace w^0 by w. Otherwise let w^0 unchanged.

Step 3. Repeat step 2, m times.

Step 4. Check that w^0 is a local optimum by comparing its utility to the utility of all its neighbors. If not, go back to step 2.

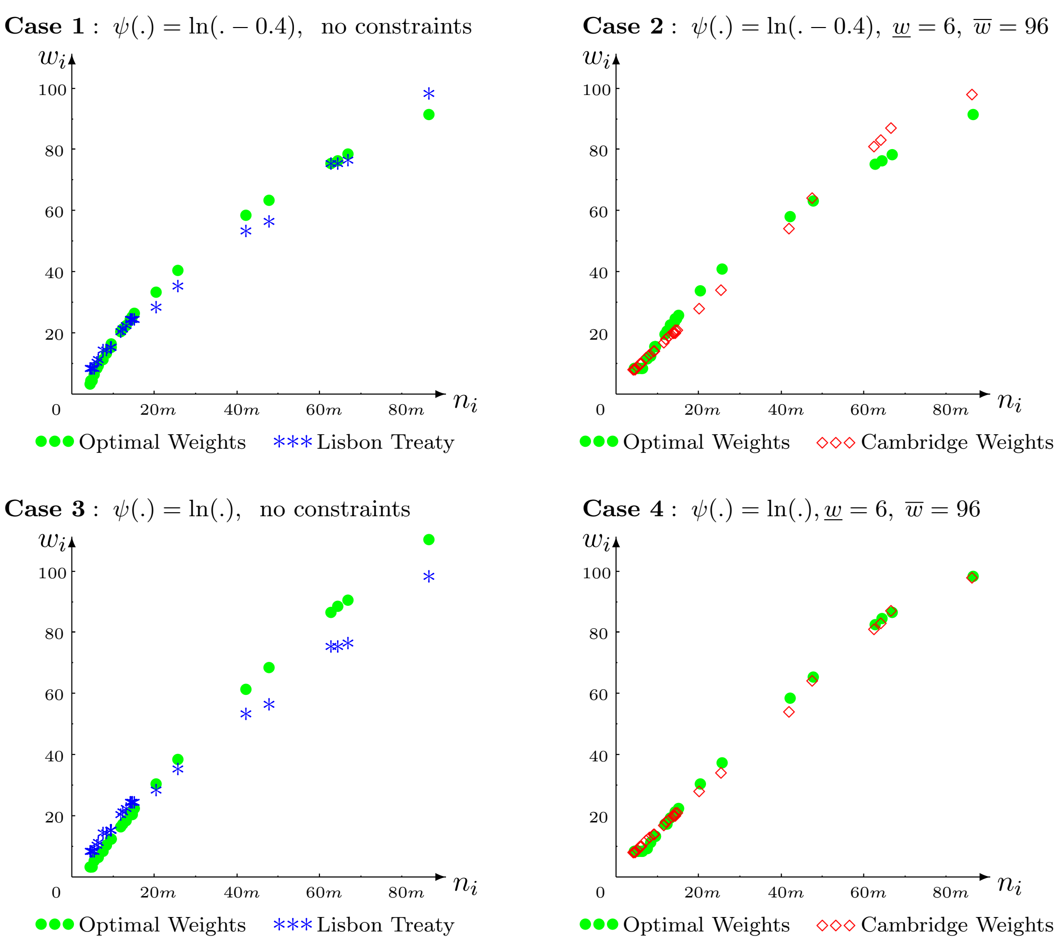
Starting from an initial random vector (step 1), the algorithm looks for neighboring improvements, m consecutive times (step 2-3). Once this is done, it checks that the obtained vector is indeed locally optimal by comparing its utility to the utility of all its neighbors (step 4). If it is not, the algorithm repeat step 2, m other times, and so on. We set $m = 1000$ in the application. The maximization strategy consists in running the algorithm (Alg2) 200 times and selecting the highest utility local optimum as the global optimum. The corresponding matlab program is given in the Appendix (A3). In the next section, we use algorithm (Alg2) to find the optimal number of representatives by country at the European Parliament.

4.2 Results: Case of the European Parliament

We consider the case of the European Parliament. The weights now correspond to the number of representatives by country and the total weight $\bar{W}$ is fixed to 751. The same setting also applies to any European council in which there is only one representative by country and decisions are taken at the weighted majority. The only difference being the total weight could then be different than 751.

We set $\gamma = 1/2$, $c = 27$, $\bar{W} = 751$. As a benchmark, we use the two following utility functions: $\psi(.) = \log(. - 0.4)$ and $\psi(.) = \log(.)$. The first one exhibits a higher degree of concavity. We consider both the case where the weights are unconstrained and case where the weights are constrained to be between 6 and 96. These constraints are the ones imposed by the Treaty of Lisbon (Article 9.A). We will refer to them as the "Lisbon constraints". They were taken into account by the Cambridge Commission, a group of experts who recently met to discuss the issue of optimal apportionment at the European Parliament. Their recommendations are summarized in ([7]).
For each of these four cases, we apply algorithm (Alg2) from 200 randomly drawn vectors of weights (generated with the uniform distribution by algorithm (Alg1)). In the four cases considered, the global optimum (or highest utility local optimum) was attained for more than 99% of the initial vectors. In case 1 ($\psi(.) = log(. - 0.4)$, no constraint), it was attained for 199 of the 200 initial vectors. These results give us further confidence that the highest utility local optimum actually corresponds to the true global optimum. The solutions are given in Figure 3. The optimal weights are pictured in green circles, the weights under the Treaty of Lisbon are pictured in blue stars, and the weights recommended by the Cambridge Commission are pictured in red diamonds.

Figure 3. *Optimal weights for the European Parliament ($c = 27$, $\bar{W} = 751$, $\gamma = 1/2$).*



Because of the restriction to integer weights, the optimal weights in cases 1 and 3 do not satisfy perfect

degressive proportionality (defined as the property that the ratio of weights to the population size decreases with the population size). However, in both cases, we verify that there exists a neighboring vector of real weights w' (such that $|w_i^* - w_i'| < 1$, $\forall i \in C$) which satisfies perfect degressive proportionality. Graphically, we observe that the curves supporting the optimal weights (solution curves) exhibit a concave shape. We conclude that the optimal weights do exhibit some form of degressive proportionality.
When the weights are constrained to be between 6 and 96 (cases 2 and 4), the solution curves take a similar shape than when unconstrained, except that they are truncated at the lower end. For $\psi(.) = \ln(. - 0.4)$, all the countries who received a weight strictly smaller than 8 in the unconstrained case are now given the minimum weight of 6. Similarly, for $\psi(.) = \ln(.)$, all the countries who received a weight strictly smaller than in the unconstrained case are now given the minimum weight of 6. In both cases, the corresponding curves exhibits a kink at the transition between the constant part and the concave part.
We note in all the cases we have considered, middle sized countries (Spain, Poland, Romania and the Netherlands) are given a bigger weight than under the treaty of Lisbon, while smaller countries (in particular Ireland, Lithuania, Latvia and Slovenia) are given a smaller (or equal) weight, even when the optimization is constrained.
Graphically, we observe the concavity of the solutions curves is stronger for $\psi(.) = ln(. - 0.4)$ than for $\psi(.) = ln(.)$, which means that the corresponding optimal weights exhibit a stronger degressive proportionality. In the following subsection, we show there is in fact a robust increasing relation between the concavity of the solution curve and concavity of the individual utility function. The optimal vectors of weights for cases 1 to 4 are presented in Appendix B.

4.3 Degressive Proportionality

In order to measure the concavity of the solution curve, we first approximate it to the closest function in the family $\left\{W_{\alpha,\beta} : n \mapsto \alpha n^{\beta}\right\}_{\alpha\in\mathbb{R},\ \beta\in[0,1]}$. Then, if the fit between the two curves is satisfactory (see below), we interpret the parameter β^* of the approximating function as a measure of the curve's concavity. This parameter can also be interpreted as the "degree of proportionality" of the corresponding optimal weights (see Bovens and Hartmann 2008 [3]) : it is equal to 0 for the constant weights and to 1 for the proportional weights.
The approximating function W_{α^*,β^*} is the one which minimizes the maximum distance to the solution curve. For a given solution curve $(n_k, w_k^*)_{k=1\dots c}$, the coefficients (α^*, β^*) are thus given by:

$$(\alpha^*, \beta^*) = \underset{(\alpha,\beta)\in\mathbb{R}_+^2}{\arg\min} \left(\max_{k=1\dots c} |w_k^* - \alpha n_k^{\beta}| \right)$$

The solution curve is concave if β^* is less than 1 and the concavity increases as β^* decreases. The corresponding matlab program is given in the Appendix (A4). We define the error as the maximum distance between the solution curve and its approximation:

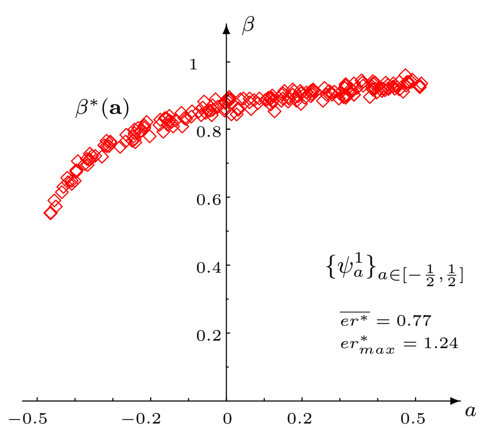
$$er^* = \max_{k=1\dots c} |w_k^* - \alpha^* n_k^{\beta^*}|$$

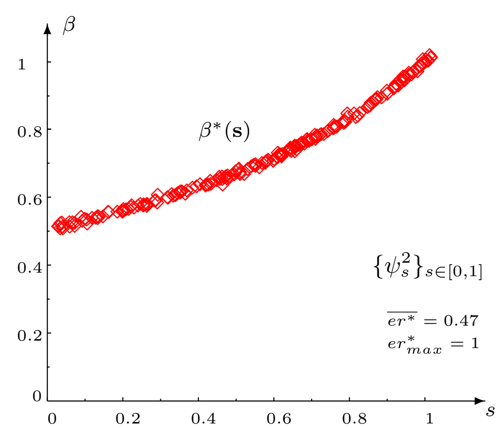
The error is measured in number of weights. We will consider that the approximation, and the corresponding measure of concavity, is satisfactory whenever the error is less than 2. This choice is arbitrary, however, given the number of countries (27) and total number of weights (751), we believe that it is enough to make sure that the approximating curve will have a similar shape than the solution curve. In the two unconstrained cases considered above, the estimated coefficients β^* were equal to 0.67 for $\psi = ln(\cdot - 0.4)$ ($er^* = 1.9$) and 0.84 for $\psi = ln(\cdot)$ ($er^* = 1.6$). These results confirm our previous observation that the concavity of the solution curve is higher for $\psi = ln(\cdot - 0.4)$ than for $\psi = ln(\cdot)$. We would like to investigate the existence of an increasing relation between the concavity of the individual utility function and concavity of the solution curve. We restrict ourselves to the two following classes of individual utility functions:

$$\psi_a^1(x) = log(x + a) \quad \text{for } a \in]-\frac{1}{2}, \frac{1}{2}]$$
$$\psi_s^2(x) = (x - \frac{1}{2})^s \quad \text{for } s \in [0, 1]$$

These functions are all smooth and well defined over our interval of interest $[\frac{1}{2}, 1]$. The coefficients a and s reflect their concavity : the lower a and s, the more concave are the corresponding utility functions ψ_a^1 and ψ_s^2. In order to estimate the variation of β^* as a function of a, we draw m random vectors of population $(n^1, ..., n^m)$ from a Zipf's law[7] and m random parameters $(a^1, ..., a^m)$ from the uniform distribution over $]-\frac{1}{2}, \frac{1}{2}]$. For each pair (n^i, a^i), $i \in \{1, \dots, m\}$, we determine the solution curve $(n_k^i, w_k^*(n^i, a^i))_{k=1\dots c}$ and then compute the corresponding coefficient β_i^*. We do the same thing for the second class of utility functions $\{\psi_s^2\}_{s\in[0,1]}$. For $m = 200$, we obtain the following results :

Figure 5. *Concavity of the solution curves ($m = 200$ points, 25 countries, $\bar{W} = 500$ seats).*





We chose to draw random vectors of population to make sure our results were not particular to the current distribution of population in the European Union. However, we do not get a thin line because the optimal weights are not fully independent of the particular choice for the vector of population. Nonetheless, we can remark that the variability of β^* for a given value of a is not very large. Overall, we observe a very clear increasing relation between the estimated coefficient β^* and the utility parameters a and s. Over the sample of 200 solution curves, the parameter β^* was measured with an average error of 0.77 (0.47) and a maximum error of 1.24 (1.00) for the ψ_a^1 (ψ_s^2) functions.

Looking at the optimal vectors of weights for the different values of a, we note while the maximum threshold of 96 weights is respected whenever a is smaller than 0.34, the minimum threshold of 6

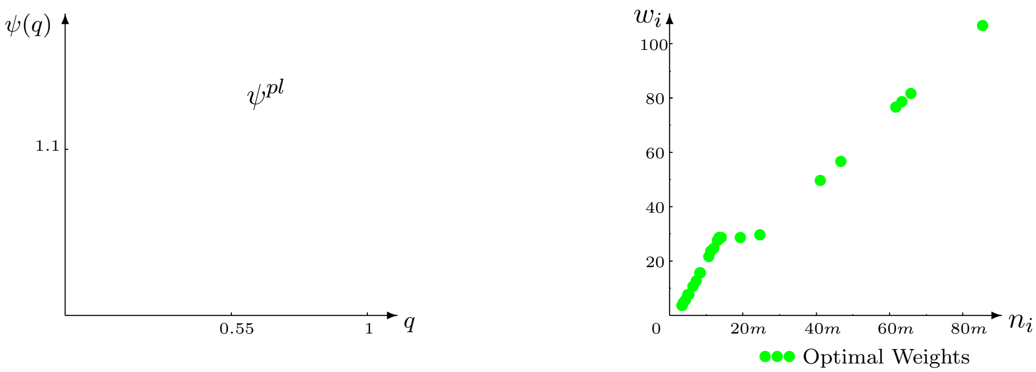
[7] $\forall k = 1 \dots 25,\ n_k \sim L,$ where $\forall n \in [100, 1000]$, $\mathbb{P}(L \geq n) = (\frac{100}{n} - 0.1)$. The Zipf's laws are well known for being a good approximation of the actual distributions of cities' or countries' sizes (Rose [14] 2006).

weights is never satisfied. A weaker minimum threshold of 5 seats is respected, but only for extremely low values of a (≈ -0.5).
We have focused here on the concavity of the solution curve. However, the solution could exhibit degressive proportionality without being concave (the concavity requirement is stronger). We give such an example below.

$$\psi^{pl} : \left| \begin{array}{ccl} [0,1] & \to & \mathbb{R} \\ x & \mapsto & \begin{cases} \psi^{pl}(x) = 2x & \text{if } x \leq 0.55 \\ \psi^{pl}(x) = 1.1 + (x - 0.55) & \text{if } x > 0.55 \end{cases} \end{array} \right.$$

Figure 6. *A counterexample with non-concave optimal weights.*

Case 5 : $\psi(.) = \psi^{pl}$, no constraints



The intuition for this particular shape is that the discontinuity in the derivative of ψ (for $q = 0.55$) creates a point of attraction for the probabilities of success. As a result, various countries of different sizes are given the same weight at the optimum. These countries have a probability of success equal to the value taken by q at the discontinuity of the derivative of ψ (0.55).

4.4 Comparisons

For a given utility function ψ, the optimal vector of weights w^* maximizes the social utility U. We wonder how the other benchmark vectors of weights fare compared to that optimum (for the European Parliament). We thus consider the six following vector of weights:

- w^{LIN} is the vector of weights as proportional to the populations.
- w^P is the vector of weights as proportional to the square root of the populations (also referred to as the Penrose weights, Penrose 1947 [13])[8].
- w^{CC} is the vector of weights as recommended by the Cambridge Commission[9]
- w^{LT} is the vector of weights as chosen under the Treaty of Lisbon.
- w^{NA} is the vector of optimal weights obtained with the Normal approximation of Koriyama et al. ([8]) using our maximization algorithm.

[8]For both w^{LIN} and w^P, the weight of each country was rounded down and the remaining seats were allocated to the countries that lost the most in the truncation.

[9]They are defined by the following formula: $w_i^{CC} = min(6 + [\frac{n_i}{d}], 96)$ where d is a divisor such that $\sum_{i \in C} w_i^{CC} = 751$. See Grimmett 2011 [7].

- w^{*C} is the optimal vector of weights when the weights are constrained to be between 6 and 96.

We evaluate these vectors with respect to both their social utility U and their respective distances to the optimum vector of weights w^*. For this purpose, we introduce the following distance measure between two vectors of weights, defined as the sum of the individual absolute differences of weights between the two vectors:

$$d(w, w') = \sum_{i \in C} |w_i - w'_i|.$$

If we interpret the set of non-decreasing vectors of weights $\Gamma^+(c, \bar{W})$ as a graph, where two vectors are linked if they are neighbors (as introduced in Definition 2), then $d(w, w')$ becomes the distance between w and w' on the graph. The distance $d(w, w^*)$ is also the minimum number of iterations for our maximization algorithm to converge to the optimal vector of weights w^* when initialized at vector w. Another interpretation of the distance is that $d(w, w')/c$ measures the absolute mean difference of weights for a country between w and w'. We could also have considered the l^2 distance, but preferred to use only the l^1 as it is easier to interpret.
For each of the two considered utility functions ($\psi = ln(. - 0.4)$ and $\psi = ln(.)$), the different vectors of weights are positioned on an horizontal axis according to their social utility. The utilities are rescaled to be comprised between 0 (for the vector generating the lowest utility) and 1 (for the optimum vector of weights). We distinguish unconstrained vectors (located on the upper axis) from vectors for which the weights are constrained to be between 6 and 96 (located on the lower axis). The vectors' distance to the optimum vector of weights, $d(w, w^*)$, is given in parentheses.

Figure 7. $\psi = ln(. - 0.4)$

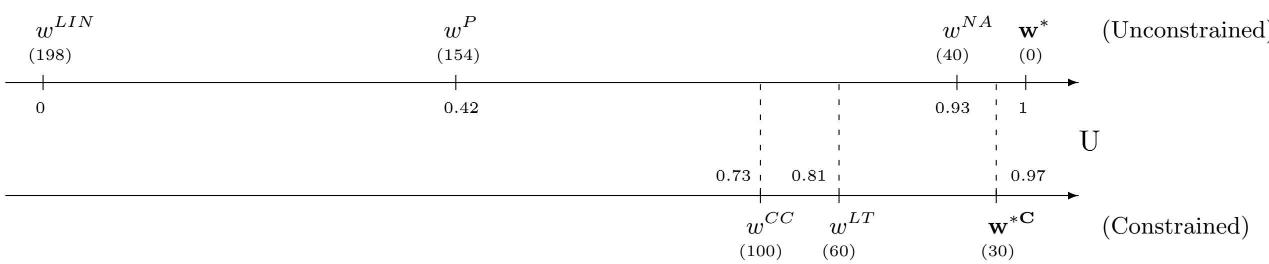
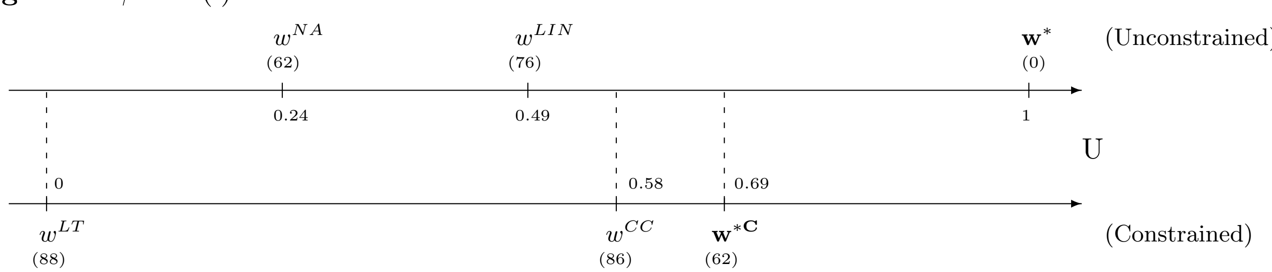


Figure 8. $\psi = ln(.)$



The utility of vector w^P (Penrose weights) being much lower than the utility of all the other vectors for $\psi = ln(.)$, we chose not to represent it in Figure 8. As expected, the respective order of the different vectors is quite sensitive to the particular choice for the utility function ψ. The approached optimum (obtained with the normal approximation) gets very close to the true optimum for $\psi = ln(. - 0.4)$, but is outperformed by three of the other benchmark vectors for $\psi = ln(.)$. These results suggest that Koriyama et al.'s normal approximation yields better results for more concave utility functions. Also,

we observe the constraints do not lead to a huge loss in utility as the constrained optimum gets very close to the actual optimum.
We conduct the same analysis for 50 other values of the parameter a indexing the class of functions $\{\psi_a^1\}_{a\in[-1/2,1/2]}$. The results are presented in the two following graphs. For each value of a, the utilities (Figure 9) were rescaled to be comprised between 0 (for the vector generating the first decile of U over $\Gamma^+(c,W)$[10]) and 1 (for the optimum vector of weights). The distances to the optimum vector of weights are represented in Figure 10.

Figure 9. *Utility comparisons for* $\psi \in \{\psi_a^1\}_{a\in[-1/2,1/2]}$.

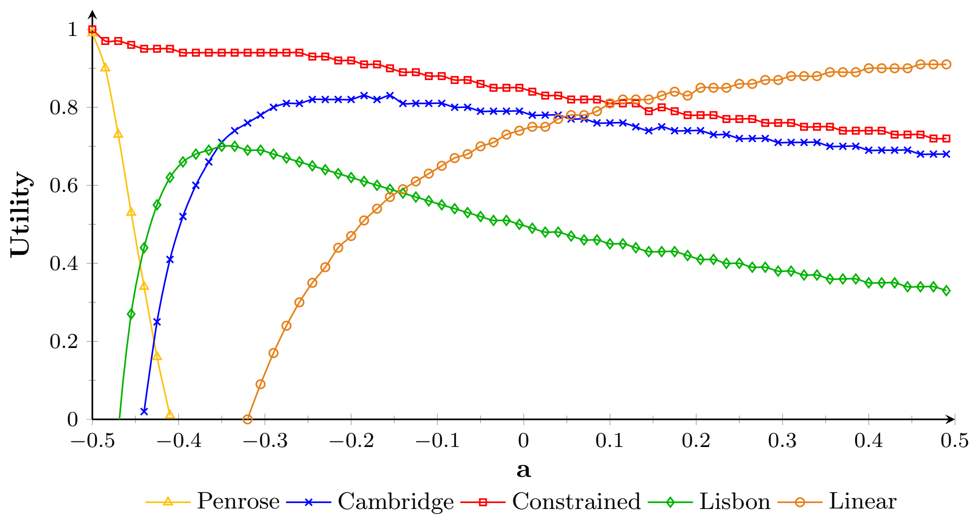
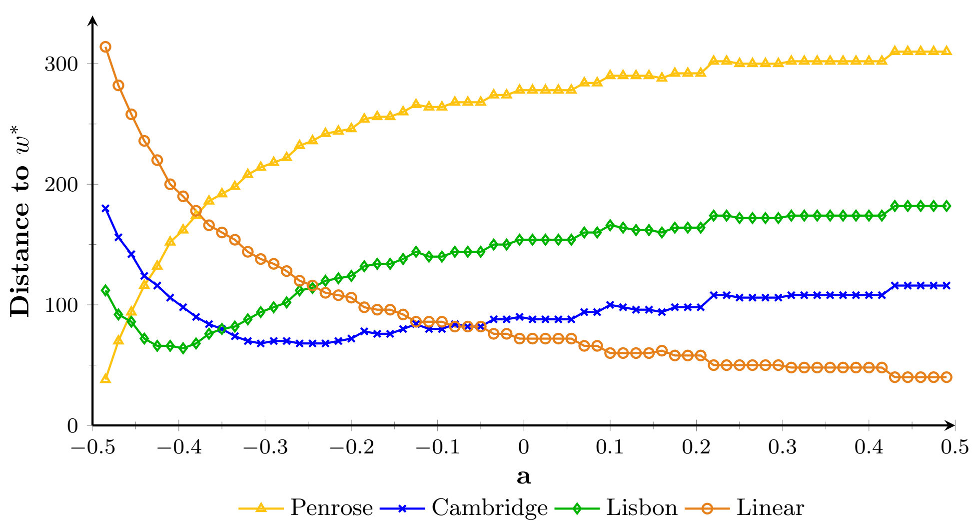


Figure 10. *Distance comparisons for* $\psi \in \{\psi_a^1\}_{a\in[-1/2,1/2]}$.



These results confirm our first impression that when the concavity of the individual utility function increases (a decreases), the constrained optimum gets closer to the unconstrained optimum (both in

[10]Measured empirically using algorithm A1.

terms of utility and distance). At the limit, when $a = -0.5$, the distance between the two optima is only of 2 weights. Therefore, we conclude that for very small values of a, the model of Koriyama et al. can almost justify the constraints imposed by the Lisbon Treaty. The Penrose weights, which perform very well for a equal to -0.5, become very inefficient when a increases. For $a > -0.4$, the utility of w^P is even lower than the first decile of U on $\Gamma^+(c, \bar{W})$, which means that there are at least 10^{21} non decreasing vectors of weights which perform better than w^P. Inversely, when a gets very close to 0.5, it is the vector of linear weights which performs very well. This is not surprising because the utility function ψ_a^1 comes close to a linear function for large values of a[11]. Also, we observe the two other benchmark vectors, the vector of weights under the Treaty of Lisbon and vector of weights recommended by the Cambridge commission, give good results as they both generate a higher utility than the first decile of U for most values of a (whenever $a > -0.44$ for w^{CC} and whenever $a > -0.47$ for w^{LT}). Furthermore, we note that the Cambridge weights, which perform in general better than the Lisbon weights ($U(w^{CC}) > U(w^{LT})$ whenever $a > -0.35$), get very close to the constrained optimum over a large interval ($a > -0.2$). These results suggest that the Cambridge weights offer a very acceptable solution to the issue of fair apportionment when the Lisbon constraints have to be satisfied.

[11]When a is big, $ln(x + a) = ln(a) + ln(1 + \frac{x}{a}) \approx ln(a) + \frac{x}{a}$, which is a linear function of x.

References

[1] Barbera S., Jackson M., 2006. “On the Weights of Nations: Assigning Voting Power to Heterogeneous Voters”, *Journal of Political Economy*, Vol. 114(2), pages 317:339.

[2] Beisbart C., Bovens L.,2007. “Welfarist evaluations of decision rules for boards of representatives”, *Social choice and Welfare*, Vol. 29, pages 581-608.

[3] Bovens L., Hartmann S., 2008. “Welfare, Voting and the Constitution of a Federal Assembly‘”, In : Galavotti M.C., Scazzieri R., Suppes P. (eds), *Reasoning, rationality and probability*, CSLI Publications, Stanford.

[4] Duff A., 2010. “Draft Report on a proposal for a modification of the Act concerning the election of the members of the European Parliament by direct universal suffrage of 20 September 1976 (2009/2134(INI))” *European Parliament, Committee on Constitutional Affairs.*

[5] Felsenthal D., Machover M., 1998. “The Measurement of Voting Power”, *Cheltentham: Edward Elgar.*

[6] Fleurbaey M., 2009. “One Stake, One Vote”, *Mimeo.*

[7] Grimmett G.R., 2011. ”European Apportionment via the Cambridge Compromise”, *arXiv:1105.4294v1.*

[8] Koriyama Y., Laslier J.F., Macé A., Treibich R., 2011. “Optimal Apportionment”, *Working paper*, sites.google.com/site/jflaslierhomepage/Home/JFL-Research.

[9] Laruelle A., Valenciano F., 2005. ”Assessing success and decisiveness in voting situations”, *Social Choice and Welfare* vol. 24(1), pages 171-197.

[10] Laruelle A., Valenciano F., 2008. “Voting and Collective Decision-Making: Bargaining and Power”, *Cambridge University Press, Cambridge, New York.*

[11] Laruelle A., Valenciano F., 2010. ”Egalitarianism and Utilitarianism in Committees of Representatives”, *Social Choice and Welfare* vol. 35, pages 221-243.

[12] Leech, D., 2002. ”Computation of Power Indices”, *Warwick Economic Research Papers*, No 644.

[13] Penrose L., 1947. “The Elementary Statistics of Majority Voting”, *Journal of the Royal Statistical Society*, vol. 109, pages 53-57.

[14] Rose A., 2006. “Cities and Countries” *Journal of Money, Credit, and Banking*, vol. 38(8), pages 2225-2245.

[15] “Treaty of Lisbon amending the Treaty on European Union and the Treaty establishing the European Community, signed at Lisbon, 13 December 2007‘”, *Official Journal of the European Union.*

5 Annex A

A1: Matlab program to compute the exact probabilities

```
function [S] = Exact(W,g)
    n=27;
    Sum=sum(W);
    S=zeros(n,1);
    s=ceil(Sum/2);
    for i=1:n
        Wb=zeros(n-1,1);
        j=0;
        for h=1:n
            if h =i
                j=j+1;
                Wb(j,1)=W(h,1);
            end
        end
        SW=sum(Wb)
        D=zeros(SW+1,n);
        D(1,1)=1;
            for r=1:n-1
                for m=SW+1:-1:1
                    for k=n:-1:1
                        if k>1 and m>Wb(r,1)
                            D(m,k)=D(m,k)
                                +D(m-Wb(r,1),k-1);
                        end
                    end
                end
            end
        p1=0;
        for l=1:n-1
            for m=s-W(i,1):SW
                p1=p1+g^l(1-g)^(n-l-1)D(m+1,l+1);
            end
         end
        p0=0;
        for l=1:n-1
            for m=s:SW
                p0=p0+g^l(1-g)^(n-l-1)D(m+1,l+1);
            end
        end
        S(i,1)=gp1+(1-g)(1-p0);
    end
end
```

A2: Matlab program to generate increasing vectors of weights

```
function[P]=generate(n,w,wmin,wmax)
P=zeros(n,1);
Test=0;
w2=w+n*(n+1)/2;
while(Test==0)
    test=0;
    for k=1:(n-1)
        X(k,1)=unidrnd(w2+1)-1;
    end
    Y=sort(X);
    for k=1:(n-2)
        if Y(k,1)==Y(k+1,1)
            test=test+1;
        end
    end
    W(1,1)=Y(1,1);
    for k=2:(n-1)
        W(k,1)=Y(k,1)-Y(k-1,1);
    end
    W(n,1)=w2-Y(n-1,1);
    W(:,2)=sort(W(:,1));
    for k=1:(n-1)
        if W(k,2)==W(k+1,2)
            test=test+10;
        end
    end
    for k=1:n
        W(k,3)=W(k,2)-k;
    end
    if W(1,3)<wmin | W(n,3)>wmax
        test=test+100;
    end
    if test==0
        Test=1;
    end
end
P=W(:,3)
end
```

A3: Matlab program to find the local optima

```
function[Smax,Umax]=Maximize(N,w,wmin,wmax)
n=length(N);
nbneighbors=n*(n-1)+1;
s=ceil(w/2);
c=1;
test=1;
while(c¡2)
    if test==1
        S=generate(n,w,wmin,wmax);
    end
    U=-10^20;
    for l=1:2000
        i=unidrnd(n);
        j=unidrnd(n);
        T=S;
        if S(i,1)>wmin & i≠j & S(j,1)<wmax
            T(i,1)=S(i,1)-1;
            T(j,1)=S(j,1)+1;
            T=sort(T);
            U2=Uexact(N,T)
            if U2>U
                S=T;
                U=U2;
            end
        end
    end
    test=1;
    K=zeros(n,nbneighbors);
    for l=1:nbneighbors
        K(:,l)=S(:,1);
    end
    l=2;
    for i=1:n
        for j=1:n
            if j≠i & K(i,1)>wmin & K(j,1)<wmax
                K(i,l)=K(i,1)-1;
                K(j,l)=K(j,1)+1;
                l=l+1;
            end
        end
    end
    for l=2:som
        u=Uexact(N,K(:,l));
        if u>U
            test=0;
        end
    end
    if test==1
        Umax(1,c)=U
        Smax(:,c)=S
        c=c+1
    end
end
Smax
Umax
end
```

%Uexact computes the utility with the program

"Exact"

A4: Matlab program to approximate the solution curve

```
function[bestb,diff]=DP(N,W)
n=length(N);
diff=W(n,1)+10;
bestk=0;
bestb=1;
for m=1:3
    for k=0:20
        b=bestb+(k-10)/10^m;
        amin=W(1,1)/N(1,1)^b;
        amax=W(1,1)/N(1,1)^b;
        for i=2:n
            ai=W(i,1)/N(i,1)^b;
            if ai<amin
                amin=ai;
            end
            if ai>amax
                amax=ai;
            end
        end
        c=criteremax(N,W,amin,b);
        nb=10^(max(m,2));
        for j=1:nb
            a=amin+(j/nb)*(amax-amin);
            d=criteremax(N,W,a,b);
            if d<c
                ab=a;
                c=d;
            end
        end
        if c<diff
            diff=c;
            bestk=k;
        end
    end
    bestb=bestb+(bestk-10)/10^m;
end
end
```

% criteremax(N,W,a,b)=$\max_{k=1\ldots n} |W_k - aN_k^b|$

6 Appendix B

Table 2: Optimal Weights.

Countries	Population	Lisbon	Cambridge	Case 1	Case 2	Case 3	Case 4
Germany	82.4 M	96	96	89	88	108	96
France	62.9 M	74	85	76	75	88	84
United Kingdom	60.4 M	73	81	74	73	86	82
Italy	58.7 M	73	79	73	72	84	80
Spain	43.7 M	54	62	61	60	66	63
Poland	38.2 M	51	52	56	55	59	56
Romania	21.6 M	33	32	38	38	36	35
Netherlands	16.3 M	26	26	31	31	28	28
Greece	11.1 M	22	19	24	23	20	20
Portugal	10.6 M	22	19	23	22	19	19
Belgium	10.5 M	22	18	23	22	18	19
Czech Republic	10.3 M	22	18	22	22	18	19
Hungary	10.1 M	22	18	22	21	18	18
Sweden	9.0 M	20	17	20	20	16	17
Austria	8.3 M	19	16	19	18	15	15
Bulgaria	7.8 M	18	15	18	17	14	15
Denmark	5.4 M	13	12	14	13	10	11
Slovak Republic	5.4 M	13	12	13	13	10	11
Finland	5.3 M	13	12	13	13	10	11
Ireland	4.2 M	12	11	11	10	8	9
Lithuania	3.4 M	12	10	9	9	6	7
Latvia	2.9 M	9	8	7	6	4	6
Slovenia	2.0 M	8	8	6	6	4	6
Estonia	1.3 M	6	7	4	6	3	6
Cyprus	0.8 M	6	6	2	6	1	6
Luxembourg	0.5 M	6	6	2	6	1	6
Malta	0.4 M	6	6	1	6	1	6
Total	492.9 M	751	751	751	751	751	751

Lisbon: Number of weights under the Treaty of Lisbon.
Cambridge: Recommendation of the Cambridge Commission.
Case 1: Optimum for $\psi(.) = \ln(. - 0.4)$, no constraints.
Case 2: Optimum for $\psi(.) = \ln(. - 0.4)$, $\underline{w} = 6$, $\overline{w} = 96$.
Case 3: Optimum for $\psi(.) = \ln(.)$, no constraints.
Case 4: Optimum for $\psi(.) = \ln(.)$, $\underline{w} = 6$, $\overline{w} = 96$.