On the convergence of the Generalized ibn Ezra Value

Louis de Mesnard

Univ. Bourgogne Franche-Comté, CREGO (EA 7317).
IAE Dijon, 2 Bd Gabriel, 21000 DIJON, FRANCE.
E-mail: louis.de-mesnard@u-bourgogne.fr

Abstract

Ibn Ezra (ibn Ezra 1146; Rabinovitch 1973; O'Neill 1982) proposed a method for solving the "rights arbitration problem" (one of the historical problems of "bankruptcy") for n claimants when the estate E is equal to the largest claim. However, when the greatest claim is for less than the estate, the question of what to do with the difference between E and the largest claim is posed. Alcalde et al.'s (2005) Generalized ibn Ezra Value (GiEV), solves the problem in T iterations, of n steps.

By using Monte-Carlo experiments, we show that: (i) T grows linearly with the number of claimants, which makes GiEV rapidly impracticable for real applications. (ii) The more E is close to the total claim d, the more T grows: T linearly grows when E exponentially approaches d by a factor 10. Moreover, we proved through theory that GiEV fails to provide a solution in a finite number of iterations for the trivial case E = d, whereas it should obviously find a solution in one iteration. So, even if GiEV is convergent, the sum of claims d appears as an asymptote: the number of iterations tends to infinite when the estate E approaches the claims total d. We conclude that GiEV is inefficient and usable only when: (1) the number of claimants is low, and (2) the estate E is largely lower than the total claims d.

JEL classification. D31, D63, D71, B1, B4

Keywords. Game theory; ibn Ezra; bankruptcy; rights arbitration; cooperative game; convergence; Monte-Carlo experiments.

Running head. Convergence of the Generalized ibn Ezra Value.

1 Introduction

In an allocation problem, when the estate to be shared E is larger than the sum of claims d, each claimant is obviously served without any rationing. The allocation problem, the so-called "bankruptcy problem", begins when the value of the estate amounts to less than the sum of the claims. When the estate to be shared E is just equal to the sum of the claims d, i.e., $E = d$, the whole estate can be allocated without any competition between the claimants, each one trivially receiving exactly what he claims. Ibn Ezra (ibn Ezra 1146; Rabinovitch 1973; O'Neill 1982; Bergantiños and Méndez-Naya 2001; Chun and Thomson 2005; Alcalde et al. 2008) proposed a method for solving the bankruptcy problem that he called *the rights arbitration problem*[1] where the estate E is equal to the greatest claim of n claimants. As expounded by O'Neill (1982), it can be solved in n steps and poses no difficulty. However, when the maximum claim is for less than the available estate, the question is: what do we do with the difference between the estate and the largest claim? Who receives it? A number of solutions are possible. The most popular is O'Neill's (1982) *Minimal Overlap Rule*.[2] Alcade et al. (2005) criticize this procedure because it mixes two principles of equity: "Up to a certain amount of estate we should follow the recommendations by Ibn Ezra and, after it, we should divide divide the extra estate trying to equalize agents' loses" (Alcalde et al. 2005, p. 15).[3] This is why they propose another attractive solution, the *Generalized ibn Ezra Value*, by imposing "that the general principle in which the recommendations by this author [ibn Ezra] are inspired should remain fixed" (Alcalde et al. 2005, p. 15). This method is iterative, that is, we have T iterations (each one including the n steps of ibn Ezra - O'Neill). In this paper, we examine the computational and mathematical properties of the algorithm of the *Generalized ibn Ezra Value* to explain why the method poses problems.

2 Ibn Ezra's problem in formalized terms

A bankruptcy problem is defined as follows. Consider a finite set $N = \{1, ..., n\}$ of claimants or creditors (with $|N| = n$), a vector $\mathbf{d} \in \mathbb{R}^n$ of claims ordered in increasing order, i.e., $d_1 \leq d_2 \leq ... \leq d_n$; we denote $d \equiv \sum_{i=1}^{n} d_i$ and consider an estate to be distributed $E \in \mathbb{R}_+$. The resource is scarce: $E \leq d$. The solution is the payoff vector $\mathbf{x} \in \mathbb{R}^n$. By denoting $x \equiv \sum_{i=1}^{n} x_i$, the solution is such that the following axioms are fulfilled:[4]

1. Axiom 1. $0 \leq x_i \leq d_i$ for any $i \in N$: no one can receive more than he claims.

[1] It is one of the historical problems of "bankruptcy", posed years ago.

[2] See also Chun and Thomson (2005) or Alcalde et al. (2008). The *Minimal Overlap Rule* is also called *Minimal Overlap Value*.

[3] This second rule is the so-called Constrained Equal Loss Rule.

[4] For more about the axiomatic approach to bankruptcy problems, see Peyton Young (1987) or Thomson (2003).

2. Axiom 2. $x = E$.

We suppose that $d \geq E$, where $d \equiv \sum_{i=1}^{n} d_i$, that is, the problem is a "rationing" problem. For $E > d$, the problem is trivial: it is no longer a bankruptcy game because each claimant receives what he/she claims.

Remark. For $E = d$, any allocation problems turn out to be trivial: they turn into a simple division, with each claimant receiving exactly what he claims, that is, $x_i = d_i$ for any i because each claimant can be served exactly without any competition among claimants.

Moreover, to prevent either claimant from beating the other by claiming an infinite amount, the claims are truncated (Bergantiños and Méndez-Naya 2001, p. 225; Moulin 2003, p. 37–38, 262), that is, replaced by E when $d_i > E$: the claims turn out to be $\tilde{d}_i = \min(d_i, E)$ for any i. The truncated claims are ordered such that $\tilde{d}_1 \leq \tilde{d}_2 \leq ... \leq \tilde{d}_n$ with $\tilde{d}_n \leq E$. Obviously, $E \leq \tilde{d}$. In what follows, we omit the tilde to alleviate the notations.

As explained by O'Neill (1982), when $d_n = E$, the procedure runs as follows. At step 1, each of the n claimants receives $\frac{d_1}{n}$; the procedure stops for claimant 1. At step 2, if $d_1 = E$, the procedure stops for good. Otherwise, if $d_1 < E$, each claimant receives $\frac{d_1}{n} + \frac{d_2 - d_1}{n-1}$ and the procedure stops for claimant 2 who departs. And so on. This continues until step n where $d_n = E$. A generic claimant i receives (after positing $d_0 \equiv 0$):

$$x_i = \sum_{j=1}^{i} \frac{d_j - d_{j-1}}{n - j + 1} \text{ for any } i = 1, ..., n \tag{1}$$

We observe that

$$x_i = x_{i-1} + \frac{d_i - d_{i-1}}{n - i + 1} \text{ for any } i = 2, ..., n$$

Table 1 shows that the total distributed at the end of any step i is equal to the total of the previous step $i - 1$ plus what is distributed at step i. Overall, only d_n is distributed in all steps. This is not a problem here as $d_n = E$ and the whole estate is distributed. When $d_n > E$, d_n is truncated and replaced by E and we retrieve the preceding case.

	Step j						
Claimant i	1	2	3	4	...	n	x_i
1	$\frac{d_1}{n}$			...	...		$\frac{d_1}{n}$
2	$\frac{d_1}{n}$	$\frac{d_2-d_1}{n-1}$		...	...		$x_1 + \frac{d_2-d_1}{n-1}$
3	$\frac{d_1}{n}$	$\frac{d_2-d_1}{n-1}$	$\frac{d_3-d_2}{n-2}$	...	...		$x_2 + \frac{d_3-d_2}{n-2}$
4	$\frac{d_1}{n}$	$\frac{d_2-d_1}{n-1}$	$\frac{d_3-d_2}{n-2}$	$\frac{d_4-d_3}{n-3}$	...		$x_3 + \frac{d_4-d_3}{n-3}$
...	...	...	...	...	...	...	...
n	$\frac{d_1}{n}$	$\frac{d_2-d_1}{n-1}$	$\frac{d_3-d_2}{n-2}$	$\frac{d_4-d_3}{n-3}$	...	$d_n - d_{n-1}$	$x_{n-1} +$ $d_n - d_{n-1}$
Total distributed during step j	d_1	$d_2 - d_1$	$d_3 - d_2$	$d_4 - d_3$	...	$d_n - d_{n-1}$	
Total distributed in all steps up to j	d_1	d_2	d_3	d_4	...	d_n	d_n

Table 1: Ibn Ezra procedure (case where $d_n = E$)

The difficulty arises when the greatest claim is smaller than the estate (Chun and Thomson 2005; Alcalde et al. 2008). The problem is to decide how we allocate the unallocated surplus $E - d_n$. A greatest claim smaller than the estate, i.e., $d_n < E$, should be a possibility, even if it is ignored by ibn Ezra in the historical rights arbitration problem.

3 Alcalde et al.'s Generalized ibn Ezra Value

When $d_n < E$, Alcalde et al.'s (2005) *Generalized ibn Ezra Value* is based on the following principle: at each iteration t, the ibn Ezra procedure is applied, that is, each claimant i receives what claimant $i-1$ has already received, plus

$$\left(d_i^t - d_{i-1}^t\right) / (n - i + 1)$$

following Table 1. Obviously, at iteration 1, the demands are $d_i^{(1)} = d_i$ for any i and claimant 1 receives d_1/n. If there is anything left to distribute out of the estate, the same process is repeated in a new iteration except the demand of each claimant i is reduced by what has already been distributed to i: $d_i^{(t+1)} = d_i^{(t)} - x_i^{(t)}$. Alcalde et al. (2005) provide the following example:

Example 1. Let $N = \{1, 2, 3\}$, $\mathbf{d} = (18, 22, 24)$, $d = 64$ and $E = 41$. At iteration 1, by applying ibn Ezra's procedure we obtain:
$\mathbf{x}^{(1)} = \left(\frac{18}{3} = 6, 6 + \frac{22-18}{2} = 8, 8 + \frac{24-22}{1} = 10\right)$ with $x^{(1)} = 24$; thus, $d_1^{(2)} = 18 - 6 = 12$, $d_2^{(2)} = 28 - 8 = 14$, $d_3^{(2)} = 24 - 10 = 14$, $d^{(2)} = 40$, and the residual

estate is $E^{(2)} = 41 - 24 = 17$.
At iteration 2, by applying ibn Ezra's procedure again we obtain:
$\mathbf{x}^{(2)} = \left(\frac{12}{3} = 4, 4 + \frac{14-12}{2} = 5, 5 + \frac{14-14}{1} = 5\right)$ with $x^{(2)} = 14$; thus, $\mathbf{d}^{(3)} = (8, 9, 9)$, $d^{(3)} = 26$, and $E^{(3)} = 3$.
At iteration 3, 1 is allocated to each claimant, again by ibn Ezra's procedure, which yields the solution $\mathbf{x} = (11, 14, 16)$.

Therefore Alcalde et al.'s (2005) method amounts to applying ibn Ezra's procedure successively on what remains to be shared. The residual estate is never increasing. Alcalde et al. (2005, pp. 37-38) prove by a property of finite convergence that the procedure converges:

Proposition 1. *The Generalized ibn Ezra Value converges: $\exists T \in \mathbb{N}/E^{(T)} = 0$ (Alcalde et al.'s (2005, p. 18).*

Proof. The proof is in Alcalde et al. (2005, pp. 37-38). □

The procedure satisfies axioms 1 and 2.

4 Convergence: Numerical approach

Alcalde et al.'s (2005) *Generalized ibn Ezra Value* is clearly attractive. However, even if convergence of the procedure is guaranteed, it could be very slow computationally as we allocate to the total only $\max_i d_i^{(t)}$ at each iteration t because the total distributed by ibn Ezra's procedure is $\sum_{i=1}^{n} x_i = d_n$ when $d_n < E$, as seen before.

Example 2. Consider the following example: $\mathbf{d} = (4, 7, 9, 10)$, $d = 30$ and $E = 26$. We apply Alcalde et al.'s (2005) *Generalized ibn Ezra Value*. Six iterations will be necessary.
At iteration 1, $\mathbf{x}^{(1)} = \left(\frac{4}{4} = 1, 1 + \frac{7-4}{3} = 2, 2 + \frac{9-7}{2} = 3, 3 + \frac{4-3}{1} = 4\right)$ with $x^{(1)} = 10$; thus, $\mathbf{d}^{(2)} = (3, 5, 6, 6)$, $d^{(2)} = 20$, and $E^{(2)} = 26 - 10 = 16$.
At iteration 2, $\mathbf{x}^{(2)} = \left(\frac{3}{4}, \frac{3}{4} + \frac{5-3}{3} = 1\frac{5}{12}, 1\frac{5}{12} + \frac{6-5}{2} = 1\frac{11}{12}, 1\frac{11}{12} + \frac{6-6}{1} = 1\frac{11}{12}\right)$ with $x^{(2)} = 6$; thus, $\mathbf{d}^{(3)} = \left(2\frac{1}{4}, 3\frac{7}{12}, 4\frac{1}{12}, 4\frac{1}{12}\right)$, $d^{(3)} = 14$, and $E^{(3)} = 16 - 6 = 10$.
At iteration 3, $\mathbf{x}^{(3)} = \left(\frac{9}{16}, 1\frac{1}{144}, 1\frac{37}{144}, 1\frac{37}{144}\right)$ with $x^{(3)} = 4\frac{1}{12}$;
thus, $\mathbf{d}^{(4)} = \left(1\frac{11}{16}, 2\frac{83}{144}, 2\frac{119}{144}, 2\frac{119}{144}\right)$, $d^{(4)} = 9\frac{11}{12}$, and $E^{(4)} = 10 - 4\frac{1}{12} = 5\frac{11}{12}$.
At iteration 4, $\mathbf{x}^{(4)} = \left(\frac{27}{64}, \frac{660}{919}, \frac{500}{593}, \frac{500}{593}\right)$ with $x^{(4)} = 2\frac{119}{144}$;
thus, $\mathbf{d}^{(5)} = \left(1\frac{17}{64}, 1\frac{684}{797}, 1\frac{703}{715}, 1\frac{703}{715}\right)$, $d^{(5)} = 7\frac{13}{144}$, and $E^{(5)} = 5\frac{11}{12} - 2\frac{119}{144} = 3\frac{13}{144}$.
At iteration 5, $\mathbf{x}^{(5)} = \left(\frac{81}{256}, \frac{295}{574}, \frac{411}{713}, \frac{411}{713}\right)$ with $x^{(5)} = 1\frac{703}{715}$;
thus, $\mathbf{d}^{(6)} = \left(\frac{243}{256}, 1\frac{304}{883}, 1\frac{24}{59}, 1\frac{24}{59}\right)$, $d^{(6)} = 5\frac{87}{439}$, and $E^{(6)} = 3\frac{13}{144} - 1\frac{703}{715} = 1\frac{47}{439}$.
At iteration 6, $\mathbf{x}^{(6)} = \left(\frac{14}{59}, \frac{69}{238}, \frac{69}{238}, \frac{69}{238}\right)$ with $x^{(6)} = 1\frac{47}{439}$;
thus, $\mathbf{d}^{(7)} = (0, 0, 0, 0)$, $d^7 = 0$, and $E^{(7)} = 1\frac{47}{439} - 1\frac{47}{439} = 0$.
The solution is $\mathbf{x} = \left(\frac{3104}{361}, \frac{5748}{791} \frac{7529}{599}, \frac{8529}{599}\right) \sim (3.29, 5.95, 7.89, 8.89)$ with $x = 26$.

So, depending on the value of E, the claims being unchanged, we may explore the change in the number of iterations that are necessary for convergence. [5] The results are depicted in Figure 1. This figure shows that the method tends to converge in a growing number of iterations when $E \to d$. We may even suspect that the number of iterations tends toward infinity when E approaches d. Paradoxically, it is when we approach the point where allocating turns out to be trivial, i.e. $E = d$, that "the ground gives way under our feet".[6] This shows that the procedure may not be computationally efficient and convergence may be very slow as the procedure may require an infinite number of steps when we approach the point where the division problem turns out to be trivial. This questions the convergence of the *Generalized ibn Ezra Value* as stated by proposition 1. However, this intuition should be proved. This is why in what follows we illustrate it by an extended numerical example, then by an extensive numerical analysis based on Monte-Carlo experiments, and finally by a formal demonstration.

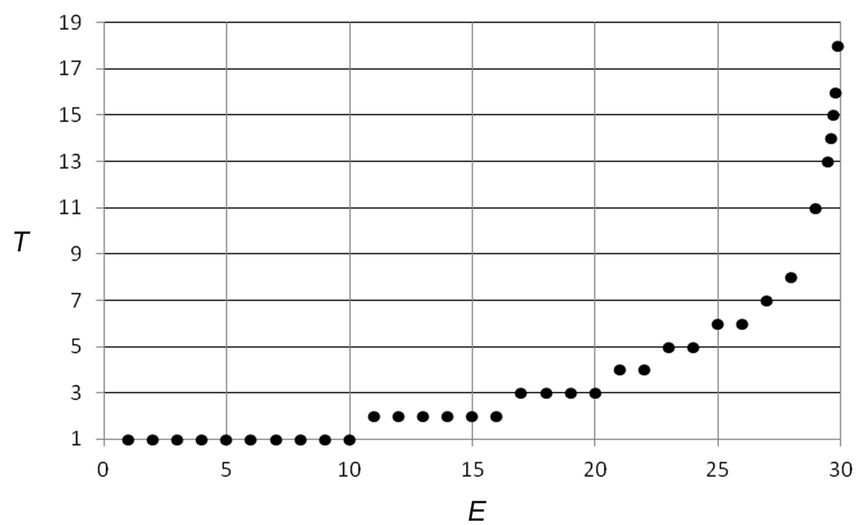

Figure 1: Alcalde et al.'s (2005) generalization: number of iterations to reach convergence, depending on E with the data of Example 2 $\mathbf{d} = (4, 7, 9, 10)$, and $d = 30$).

[5] Obviously, when $E = d_n = 10$, we have ibn Ezra's original case: the method converges in one iteration.

[6] When $E = d$ any allocation problems turn out to be trivial: they turn into a simple division, with each claimant receiving exactly what he claims, that is, $x_i = d_i$ for any i because each claimant can be served exactly without any competition among claimants.

4.1 Extension of Alcalde et al.'s example

For the extended example and the Monte-Carlo experiments, we have written a program in Maple programming language.[7] It is launched by calling the procedure $GiEV10$ shown in appendix 6. For example,

$$GiEV10(1000, 0, 0, 0, 0, 0, 0, 4, 7, 9, 10, 26)$$

launches the program for a maximum of 1000 iterations, the vector of claims $\mathbf{d} = (4, 7, 9, 10)$[8] and an estate of 26 (the data of example 2). The precision is 20 digits. The stop condition is double: either the residual estate $E^{(t)}$ is zero at iteration t, or, to avoid an infinite loop, the residual estate turns out to be constant, i.e., $E^{(t)} = E^{(t-1)}$, on the understanding that $E^{(t)}$ cannot be growing by construction, i.e., $E^{(t)} \leq E^{(t-1)}$. We have explored some extended examples ranging from four (the example 2) to 10 claimants. Table 2 indicates the claims of each claimant (in row) for each example. For example, column 6 corresponds to an example where $\mathbf{d} = (4, 7, 9, 10, 15, 16)$.

		vector**d** depending on the number of claimants						
		4	**5**	**6**	**7**	**8**	**9**	**10**
Claimant	**1**	4	4	4	4	4	4	4
	2	7	7	7	7	7	7	7
	3	9	9	9	9	9	9	9
	4	10	10	10	10	10	10	10
	5		15	15	15	15	15	15
	6			16	16	16	16	16
	7				21	21	21	21
	8					29	29	29
	9						30	30
	10							34
Total claim		30	45	61	82	111	141	175

Table 2: The seven examples with a growing number of claimants: Claims

For each of the seven examples, Table 3 gives the number of iterations T that are necessary to solve the bankruptcy problem when $E \to d$. The table should be read as follows: the example of six claimants and $E = 61 - 10^{-2} = 60.99$ requires 44 iterations to be solved. We deduce two findings of Table 3:

1. More claimants (when we read Table 3 from the left to the right) requires more iterations to solve the problem. Considering a very large number of claimants, as the whole population of a country, would "explode" the number of iterations. This could be expected because the loop for n in

[7] Exactly, Maple 18.02. Maple is a popular mathematical software scientific computing software.

[8] The procedure is able to handle up to 10 claimants. For four claimants, we set the first six claims to zero.

procedure $GiEV10$ becomes longer (i.e., we have more steps), n being multiplied by T.

2. When E grows to approach d (i.e., when we read Table 3 from the top to the bottom) we need more iterations. This result was somewhat expected because the two first columns correspond to what is depicted by Figure 1.

Examples: Number of claimants													
4		5		6		7		8		9		10	
E	T	E	T	E	T	E	T	E	T	E	T	E	T
29	11	44	15	60	20	81	24	110	29	140	35	174	40
$29-10^{-1}$	18	$45-10^{-1}$	25	$61-10^{-1}$	32	$82-10^{-1}$	41	$111-10^{-1}$	47	$141-10^{-1}$	53	$175-10^{-1}$	61
$29-10^{-2}$	26	$45-10^{-2}$	35	$61-10^{-2}$	44	$82-10^{-2}$	53	$111-10^{-2}$	62	$141-10^{-2}$	72	$175-10^{-2}$	82
$29-10^{-3}$	34	$45-10^{-3}$	46	$61-10^{-3}$	56	$82-10^{-3}$	69	$111-10^{-3}$	80	$141-10^{-3}$	91	$175-10^{-3}$	103
$29-10^{-4}$	42	$45-10^{-4}$	55	$61-10^{-4}$	70	$82-10^{-4}$	82	$111-10^{-4}$	97	$141-10^{-4}$	110	$175-10^{-4}$	125
$29-10^{-5}$	51	$45-10^{-5}$	67	$61-10^{-5}$	81	$82-10^{-5}$	98	$111-10^{-5}$	113	$141-10^{-5}$	129	$175-10^{-5}$	147
$29-10^{-6}$	59	$45-10^{-6}$	76	$61-10^{-6}$	94	$82-10^{-6}$	113	$111-10^{-6}$	129	$141-10^{-6}$	149	$175-10^{-6}$	169
$29-10^{-7}$	67	$45-10^{-7}$	86	$61-10^{-7}$	106	$82-10^{-7}$	135	$111-10^{-7}$	142	$141-10^{-7}$	168	$175-10^{-7}$	***371***
$29-10^{-8}$	75	$45-10^{-8}$	100	$61-10^{-8}$	116	$82-10^{-8}$	256	$111-10^{-8}$	146	$141-10^{-8}$	195	$175-10^{-8}$	***349***
$29-10^{-9}$	148	$45-10^{-9}$	189	$61-10^{-9}$	121	$82-10^{-9}$	256	$111-10^{-9}$	146	$141-10^{-9}$	195	$175-10^{-9}$	***349***

Table 3: The seven examples with a growing number of claimants: Number of iterations T depending on the estate E

However, such examples may be considered as particular and arbitrary. This is why we now conduct some Monte-Carlo-like experiments to confirm or infirm the intuition of the seven examples of Table 2.

4.2 Monte-Carlo experiments

We consider the gap between E and d, so that the gap is independent of the magnitude of E and d. To do this, we conduct the experiments such that the relative gap between d and E decreases exponentially in base 10, that is,

$$\frac{d-E}{d} = 10^{-g}$$

i.e., $E = (1 - 10^{-g})\,d$ (we take g integer): the parameter g is the exponent of the relative gap between d and E: $\frac{d-E}{d} = 10^{-g}$. We adapt the procedure $GiEV10$ so that it turns out to be a subroutine $GiEVr$, which is called by a new procedure, $GiEVMC$ the arguments of which are: N the maximum number of claimants, G the maximum value for g, and MC the number of times that we compute each solution of a *Generalized ibn Ezra Value* problem, with the claims being randomly chosen. The program $GiEVMC$ and the subroutine $GiEVr$ are provided in appendix 6. In $GiEVr$ we have an instruction "$E = ds\,(1 - gap)$:", where ds is for the variable d, and gap is for 10^{-g}, as transmitted by the procedure $GiEVMC$. To produce the vector $\mathbf{d}$, the demand of claimant i is deduced from the demand of claimant $i-1$ by adding a number randomly chosen between 0 and 10 as done by the instruction "$d\,[i] := d\,[i-1] + r\,()$:". Each problem converges in T iterations, a number transmitted by the subroutine $GiEVr$ to the program $GiEVMC$ (T is set "global" by the instruction "**global** T :"). The subroutine $GiEVr$ is repeated MC times and we calculate the average number of iterations necessary to reach convergence.

For example, $GiEVMC(40, 9, 100)$ computes the average number of iterations that are necessary to converge for 100 experiments, conducted for a number of claimants ranging from two to 40 and for $g = 1 \Rightarrow \frac{d-E}{d} = 10\%$ up to $g = 9 \Rightarrow \frac{d-E}{d} = 10^{-9}$. The results of $GiEVMC(40, 9, 100)$ are given in Table 4. All told, $39 \times 9 \times 100 = 35100$ ibn Ezra problems are solved by calling up the $GiEVr$ subroutine each time. As this represents a lot of computing time, we do not go further. Obviously, the average number of iterations is fractional even if the numbers of iterations are themselves integers.

					g				
	1	2	3	4	5	6	7	8	9
n = 2	2.87	6.21	9.11	11.54	14.01	17.1	20.83	23.58	27.15
3	4.94	9.73	15.25	20.79	26.19	30.24	37.11	42.91	47.1
4	6.43	13.46	20.99	28.92	35.62	43.53	51.09	59.94	66.79
5	8.02	17.15	26.4	35.8	46.15	55.87	65.91	75.67	86.98
6	9.45	20.68	32.08	43.56	55.94	68.3	80.55	92.74	104.4
7	11.12	23.95	37.6	52.37	65.69	81.1	94.66	109.77	122.98
8	13.06	27.08	43.43	59.8	76.11	92.06	109.24	125.96	142.02
9	14.43	30.91	48.72	66.9	85.52	104.72	122.17	142.09	162.78
10	15.9	34.5	54.22	73.76	94.48	116,00	136.9	157.67	179.64
11	17.47	38.4	59.08	82.02	105.08	127.83	151.31	175.01	197.96
12	19.24	41.6	65.97	89.71	114.21	141.17	165.67	190.64	216.4
13	20.54	44.65	70.93	97.79	124.88	152.29	179.47	206.05	234.48
14	22.33	48.41	76.04	105.48	133.78	163.73	192.87	222.77	252.95
15	23.98	52.4	81.95	112.53	143.54	176.22	207.78	240.3	271.13
16	25.7	55.43	87.54	120.07	153.18	187.21	221.71	256.51	290.43
17	27.32	58.4	93.18	127.6	163.32	197.88	235.74	272.19	307.94
18	28.89	62.63	98.48	134.97	173.82	210.21	249.16	287.23	328.31
19	30.4	65.31	104.11	143.48	182.42	223.12	263.56	303.19	344.52
20	32.24	69.35	110.13	150.66	192.64	235.24	277.13	321.49	363.79
21	33.68	72.94	115.38	159.15	202.33	246.63	291.15	337.11	381.07
22	35.33	76.42	120.28	164.8	211.76	260.68	306.21	352.51	401.35
23	36.74	79.96	126.2	173.79	221.55	269.3	319.76	368.55	418.19
24	38.61	82.91	132.15	181.31	230.76	281.11	334.93	385.85	436.83
25	40.17	87.52	136.74	187.31	240.54	295,00	348.56	400.64	456.93
26	41.67	90.91	142.44	195.8	249.73	306.81	360.51	416.08	476.1
27	43.12	93.98	148.37	204.34	261.02	317.64	375.95	432.84	492.54
28	44.55	97.25	153.91	211.07	270.99	329.65	389.71	450.05	511.19
29	46.35	100.89	158.76	218.7	278.44	342.87	401.59	465.01	529.6
30	48.42	104.81	164.62	225.56	289.25	353.39	417.07	483.27	546.38
31	49.76	107.93	169.77	233.73	297.5	365.34	431.3	500.37	565.38
32	51.38	112.2	175.64	241.92	308.62	377.61	446,00	517.32	584.86
33	52.61	115.43	180.41	249.66	318.53	388.94	459.64	530.2	601.74
34	54.43	118.81	186.96	256.68	327.43	401.54	473.42	546.53	622.2
35	56.06	122.25	192.94	264.56	337.07	411.19	488.69	563.48	637.67
36	57.81	125.35	197.47	271.2	347.6	422.95	501.54	581.02	658.89
37	59.08	128.78	203.64	278.73	358.25	436.94	515.85	596,00	677.61
38	60.71	132.75	208.22	286.25	366.64	448.26	527.9	611.1	695.18
39	62.21	136.04	214.14	293.76	375.04	460.09	543.29	628.06	713.52
40	64.39	139.55	220.59	301.57	386.28	471.05	554.96	645.92	732.25

Table 4: Monte-Carlo experiments: Number of iterations T depending on g such that $\frac{d-E}{d} = 10^{-g}$ and on the number of claimants n, for $g = \{1...9\}$ and $n = \{2...40\}$ claimants.

These experiments confirm the intuition of the two findings of the seven examples of Table 2 and Table 3 above. This is formalized in two dual empirical results 1 and 2 below.

Empirical result 1. In the *Generalized ibn Ezra Value*, when the gap between E and d is measured in relative terms, for a given value of $g = \lg_{10}\left(\frac{d-E}{d}\right)$, the number of iterations linearly grows with the number of claimants n.

To obtain this empirical result, we study the linear regressions

$$T = an - b \tag{2}$$

The coefficients that best fit the data of Table 4 are given in Table 5. The correlation coefficient is very good: $R^2 \geq .9999$. With the help of Table 5 we are able to forecast the results for more claimants by performing a linear regression on the coefficients a and b. We obtain

$$a = 2.1164g - .739 \text{ with } R^2 = .9995 \tag{3}$$

and

$$b = .7249g - 1.0024 \text{ with } R^2 = .9809 \tag{4}$$

For example, when $g = 9$, we may expect that $a = 18.3036$ from (3) and $b = 5.5217$ from (4), so that (2) turns out to be

$$T = 18.3036n - 5.5217 \tag{5}$$

which gives $T = 1825$ for $n = 100$. When $g = 2$ (i.e., a reasonable relative difference between E and d of 1%), (5) indicates that we should have roughly 3.5 billion iterations for 1 billion claimants (billion is a realistic number for a real bankruptcy).

g	$T = an - b$		R^2
	a	b	
1	1.603	.0371	.9999
2	3.5071	.6518	.9999
3	5.5262	1.0797	.9999
4	7.5914	1.5307	.9999
5	9.7206	2.442	.9999
6	11.888	3.2273	.9999
7	14.039	3.7605	1
8	16.252	4.9372	1
9	18.46	5.9317	1

Table 5: Monte-Carlo experiments: $T = ag - b$ depending on n

Empirical result 2. In the *Generalized ibn Ezra Value*, when the gap between E and d is measured in relative terms, for a given value n of the number of claimants, the number of iterations linearly grows with respect to $g = \lg_{10}\left(\frac{d-E}{d}\right)$. This result means that the number of iterations linearly grows each time the gap is divided by 10, which is consistent with Figure 1.

To obtain this result, we study the linear regressions

$$T = ag - b \tag{6}$$

The coefficients that best fit the data of Table 4 are given in Table 6. The correlation coefficient is very good: $R^2 \geq .9972$. Again, with the help of Table 6 we are able to forecast the results for a larger number of claimants by performing a linear regression on the coefficients a and b. We obtain

$$a = 2.1164n - .7248 \text{ with } R^2 = 1 \tag{7}$$

and

$$b = .7386n - 1.0367 \text{ with } R^2 = .9983 \tag{8}$$

For example, when $n = 100$, we may expect that $a = 210.915$ from (7) and $b = 72.823$ from (8), which implies that 6 turns out to be

$$T = 210.915g - 72.823$$

and we have $T = 1825$ for $g = 9$: we retrieve the result obtained for the projection of empirical result 1, by duality.

n	$T = ag - b$		R^2
	a	b	
2	2.9705	.1414	.9972
3	5.3558	.7503	.9886
4	7.5948	1.664	.9995
5	9.8415	2.7686	.9994
6	11.961	3.3939	.9997
7	14.129	4.0636	.9997
8	16.273	4.8344	.9997
9	18.528	6.1672	.9993
10	20.535	6.7758	.9995
11	22.701	7.4872	.9996
12	24.777	7.8172	.9996
13	26.859	8.6194	.9998
14	28.958	9.4158	.9997
15	31.128	10.101	.9996
16	33.294	11.161	.9996
17	35.321	11.763	.9996
18	37.468	12.484	.9995
19	39.478	12.932	.9997
20	41.687	13.692	.9996
21	43.685	14.042	.9997
22	46.002	14.042	.9995
23	47.903	15.734	.9997
24	50.118	16.759	.9996
25	52.296	17.765	.9995
26	54.34	18.359	.9995
27	56.379	18.582	.9997
28	58.586	19.776	.9996
29	60.587	20.465	.9995
30	62.666	20.799	.9996
31	64.908	22.197	.9995
32	67.095	22.635	.9996
33	68.976	22.974	.9997
34	71.2	24.002	.9996
35	73.138	24.143	.9996
36	75.52	26.065	.9995
37	77.64	26.547	.9996
38	79.572	27.079	.9996
39	81.766	28.144	.9995
40	83.813	28.335	.9995

Table 6: Monte-Carlo experiments: coefficients a and b of $T = ag - b$ depending on n

5 Convergence when E tends to d: Theoretical approach

A first intuition about the difficulties of convergence of the algorithm of Alcalde et al.'s (2005) is simple. Following Table 1, ibn Ezra's procedure allocates at each iteration t only $d_n^{(t)}$ to the total. As the *Generalized ibn Ezra Value* applies ibn Ezra's procedure at each iteration, it allocates a relatively minor part of the estate each time, which leads to a tardy algorithm. We will demonstrate this rigorously now. Remember that, in mathematical wording, the algorithm converges at iteration t if the residual estate $E^{(t+1)}$ is equal to zero. We begin by a simple example where $E = d$.

Example 3. We return to example 2 ($\mathbf{d} = (4, 7, 9, 10)$, $d = 30$) but we posit $E = d$, that is, $E = 30$.
At iteration 1, $\mathbf{x}^{(1)} = \left(\frac{4}{4} = 1, 1 + \frac{7-4}{3} = 2, 2 + \frac{9-7}{2} = 3, 3 + \frac{4-3}{1} = 4\right)$ with $x^{(1)} = 10$; thus, $\mathbf{d}^{(2)} = (3, 5, 6, 6)$, $d^{(2)} = 20$, and $E^{(2)} = 30 - 10 = 20$. We observe that $d^{(2)} = E^{(2)}$.
At iteration 2, $\mathbf{x}^{(2)} = \left(\frac{3}{4}, \frac{3}{4} + \frac{5-3}{3} = 1\frac{5}{12}, 1\frac{5}{12} + \frac{6-5}{2} = 1\frac{11}{12}, 1\frac{11}{12} + \frac{6-6}{1} = 1\frac{11}{12}\right)$ with $x^{(2)} = 6$; thus, $\mathbf{d}^{(3)} = \left(2\frac{1}{4}, 3\frac{7}{12}, 4\frac{1}{12}, 4\frac{1}{12}\right)$, $d^{(3)} = 14$, and $E^3 = 20 - 6 = 14$. Again $d^{(3)} = E^{(3)}$.
At iteration 3, $\mathbf{x}^{(3)} = \left(\frac{9}{16}, 1\frac{1}{99}, 1\frac{19}{74}, 1\frac{19}{74}\right)$ with $x^{(3)} = 4\frac{1}{12}$;
thus, $\mathbf{d}^{(4)} = \left(1\frac{11}{16}, 2\frac{49}{85}, 2\frac{81}{98}, 2\frac{81}{9}\right)$, $d^{(4)} = 9\frac{11}{12}$, and $E^{(4)} = 14 - 4\frac{1}{12} = 9\frac{11}{12}$ and again $d^{(4)} = E^{(4)}$.
And so on to infinity, while the result should be immediate: the trivial allocation is obviously $\mathbf{x} = (4, 7, 9, 10)$ with $x = 30$.

Lemma 1. *In the Generalized ibn Ezra Value, any iteration t for which $E^{(t)} = d^{(t)}$ does not reach the solution (i.e., $x^{(t)} = d^{(t)}$, that is, $E^{(t+1)} = 0$).*

Proof. Consider an iteration t for which $E^{(t)} = d^{(t)}$. By ibn Ezra principle, only $x^{(t)} = d_n^{(t)} < d^{(t)}$ is distributed at iteration t. There remains $E^{(t+1)} = E^{(t)} - d_n^{(t)} > 0$ to be distributed. □

Remark. Obviously, when $E < d$, there is an iteration $t = T$ for which we have $x^{(T)} = d^{(T)}$, that is, $d_n^{(t)} = d^{(t)}$, which implies $E^{(T+1)} = 0$: the algorithm converges (in six iterations in example 2), but it does so slowly as expounded in section 4.

Lemma 2. *In the Generalized ibn Ezra Value, when $E = d$, it holds that $E^{(t)} = d^{(t)}$ for every iterations.*

When $E = d$, as $E^{(t)} = d^{(t)} > 0$ for any t, the situation of iteration 1 is perpetuated indefinitely.

Proof. We will prove Lemma 2 by recurrence. We have by hypothesis n claimants and $E = d$, which implies $E - d_n = d - d_n > 0$. At the first iteration the algorithm delivers $x^{(1)} = d_n^{(1)} \equiv d_n$ to the whole set of claimants; the residual demand is $d^{(1)} = d - d_n > 0$ and a residual estate of $E^{(1)} = E - x^{(1)} = d - d_n = d^{(1)} > 0$ remains to be shared. Consequently, the residual estate is still equal to the residual demand. At iteration 2, $x^{(2)} = d_n^{(2)}$ is allocated to the whole set of claimants, so the residual demand is $d^{(2)} = d^{(1)} - d_n^{(2)} = E^{(1)} - d_n^{(2)} > 0$. The remainder to be shared is

$$E^{(2)} = E^{(1)} - x^{(2)} = d^{(1)} - d_n^{(2)} = d^{(2)} > 0$$

The residual estate is again equal to the residual demand. Let us assume that the property is true at iteration t: the quantity $x^{(t)} = d_n^{(t)}$ is allocated to the whole set of claimants, the residual demand is $d^{(t)} = d^{(t-1)} - d_n^{(t)} > 0$. This leaves

$$E^{(t)} = d^{(t)} > 0 \tag{9}$$

to be shared and we have

$$E^{(t)} - d_n^{(t+1)} > 0 \tag{10}$$

Now, we prove that the property is true at iteration $t + 1$. By the ibn Ezra procedure, the quantity $x^{(t+1)} = d_n^{(t+1)}$ is allocated to the whole set of claimants. So the residual demand is $d^{(t+1)} = d^{(t)} - d_n^{(t+1)} = E^{(t)} - d_n^{(t+1)} > 0$ by (9) and (10). There remains to be shared

$$E^{(t+1)} = E^{(t)} - x^{(t+1)} = d^{(t)} - d_n^{(t+1)} = d^{(t+1)} > 0 \tag{11}$$

The residual estate $E^{(t+1)}$is still equal to the residual demand $d^{(t+1)}$. □

Lemma 3. *In the Generalized ibn Ezra Value when $E = d$, the residual estate $E^{(t)}$ is always decreasing for any t.*

Proof. From (11), $E^{(t)} < E^{(t-1)}$ for any t. □

Theorem 1. *In the Generalized ibn Ezra Value, when $E = d$, Alcalde et al.'s (2005) algorithm does not converge in a finite number of iterations.*

Proof. Lemma 2 shows that all iterations are of the type $E^{(t)} = d^{(t)}$ and Lemma 1 shows that these iterations do not reach the solution. Combining them, we deduce that no iteration is able to reach the solution. As Lemma 3 indicates that the residual estate is always decreasing, the algorithm runs infinitely, $E^{(t)}$ asymptotically tending to zero. □

Theorem 1 means that T is infinite when $E = d$. It expounds a particularly annoying problem if we consider the case $E = d$: the procedure should yield the exact solution because the bankruptcy problem turns out to be trivial, and, as the algorithm fails to reach the solution in a finite number of iterations when $E = d$ following Theorem 1, there is a clear flaw in Alcalde et al.'s (2005) *Generalized ibn Ezra Value*.

Remark. In mathematical terms, the function $E \in \mathbb{R}^+ \to T_d(E) \in \mathbb{N}$ **is not defined** when $E = d$. d is an asymptote and the function $T_d(E)$ tends to infinity when E tends to its asymptote d.[9]

[9]To take a comparison, the function $y = \frac{1}{1-x}$ is not defined for $x = 1$ and the function tends to infinity when $x \to 1$.

6 Conclusion

We have n claimants who want to share out an estate. The total of their claims d is larger than the estate E (otherwise, we have no problem of apportionment). When the maximum claim d_n (that of the n^{th} claimant) is equal to the estate—knowing that the claims that exceed the available estate are truncated—, O'Neill (1982) explains clearly how the solution to ibn Ezra's problem can be easily found in n steps. However, when the greatest claim is for less than the estate, the question of what to do with the difference $E - d_n$ is posed. An attractive answer is that proposed by Alcalde et al. (2005): the *Generalized ibn Ezra Value*, which solves the problem in T iterations, of n steps; this algorithm is convergent (Alcalde et al. (2005, p. 18). However, this algorithm fails on two points:

- First, we have shown by numerical experiments that the number of iterations grows linearly with respect to the number of claimants n, which makes the *Generalized ibn Ezra Value* impracticable when the number of claimants is large, which is the case in most real applications (empirical result 1).

- Secondly, again by numerical experiments, we shown that the number of iterations linearly grows when the estate E tends exponentially to the total claims d: if we define by g the exponent of the relative gap between d and E (such that $\frac{d-E}{d} = 10^{-g}$, i.e., the percentage $\frac{d-E}{d}$ varies by a magnitude of 10 each time), the number of iterations grows linearly with respect to g when E approaches to d (empirical result 2). Moreover, we have proved through theory by a fundamental theorem that the *Generalized ibn Ezra Value* algorithm fails to give a solution in a finite number of iterations in the trivial case $E = d$ (theorem 1), whereas it should obviously find a solution in one iteration. Overall, if we combine these two last results, even if the *Generalized ibn Ezra Value* is convergent, the sum of claims d appears as an asymptote that can only be reached to the price of a growing number of iterations, that tends to infinite when the estate E approaches the claims total d.

We conclude that the *Generalized ibn Ezra Value* algorithm is inefficient and usable only when: (i) the number of claimants is low, **and** (ii) the estate E is largely lower than the total claims d. This singularly reduces its interest.

For future researches, it could be interesting to examine if it is the case for other methods, such that the O'Neil's (1982) *Minimal Overlap Rule* and Bergantiños and Méndez-Naya's (2001) *Extended Ibn Ezra Rule*, by examining their convergence properties in a similar manner.[10] This is another story.

[10] Beyond the replication analyses conducted by Chun and Thomson (2005) on the *Minimal Overlap Rule*.

References

ALCALDE J., MARCO M. and SILVA J. (2005) "Bankruptcy games and the Ibn Ezra's proposal," *Economic Theory*, 26,1: 103-114.

ALCALDE J., MARCO M. and SILVA J. (2008) "The minimal overlap rule revisited," *Social Choice and Welfare*, 31: 109-128.

BERGANTINOS G. and MENDEZ-NAYA L. (2001) "Additivity in bankruptcy problems and in allocation problems", *Spanish Economic Review*, 3, 3: 223-229.

CHUN, Y. and THOMSON W. (2005) "Convergence under replication of rules to adjudicate conflicting claims," *Games and Economic Behavior*, 50,2: 129-142.

Ibn EZRA A. (1146) *Sefar ha-Mispar* ("The Book of the Number," in Hebrew), Verona (German trans: Silberberg M. (1895), Frankfurt am Main: Kauffmann.

MOULIN H.J. (2003) *Fair Division and Collective Welfare*, Cambridge: MIT Press.

PEYTON YOUNG H. (1987) "On dividing an amount to individual claims or liabilities", *Mathematics of Operations Research*, 12, 3: 398-414.

O'NEILL B. (1982) "A problem of rights arbitration from the Talmud," *Mathematical Social Sciences*, 2, 4: 345-71.

RABINOVITCH N.L. (1973) *Probability and Statistical Inference in medieval Jewish Literature*. Toronto: University of Toronto Press.

THOMSON W. (2003) "Axiomatic and game-theoretic analysis of bankruptcy and taxation problems: a survey," *Mathematical Social Sciences*, 45: 249-297.

Appendix

Procedure GiEV10

```
GiEV10 := proc(tmax, d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_10, E)
d := convert([d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_10], array) :
s := convert([1, 1, 1, 1, 1, 1, 1, 1, 1, 1], array) :
n := evalm(s*s) :
ds := evalm(d*s) :
print(`vector d = `, d, `d = `, ds) :
D : array(1 ..n, 1 ..tmax) :
X : array(1 ..n, 1 ..tmax) :
x := vector(n) :
xs := 0 :
Er := E :
Ert : vector(tmax) :
for i from 1 to n do
    D[i, 1] := d[i] :
    x[i] := 0 :
od:
for t from 1 to tmax do
    X[1, t] := D[1, t]/n :
    x[1] := x[1] + X[1, t] :
    Er := Er-X[1, t] :
    for i from 2 to n do
        X[i, t] := X[i-1, t] + (D[i, t]-D[i-1, t]) / (n-i + 1) :
        x[i] := x[i] + X[i, t] :
        Er := Er-X[i, t] :
    od:
    for i from 1 to n do
        D[i, t + 1] := min(D[i, t]-X[i, t], Er) :
    od:
    Ert[t] := Er :
    if Ert[t] = Ert[t-1] then break fi:
    if Er ≤ 0 then break fi:
od:
print(`Converges for T = `, t) :
for i from 1 to n do
    xs := xs + x[i] :
    print(`x_`, i, ` = `, evalf(x[i])) :
od:
print(`x = `, evalf(xs)) :
end proc:
```

Figure 2: Procedure GiEV10

Subroutine GiEVr, procedure GiEVMC

```
GiEVr := proc(tmax, n, gap)
global T :
d := vector(n) :
r := rand(0..10) :
d[1] := r( ) :
ds := d[1] :
for i from 2 to n do
    d[i] := d[i-1] + r( ) :
    ds := ds + d[i] :
od:
E := ds·(1 − gap) :
D : array(1..n, 1..tmax) :
X : array(1..n, 1..tmax) :
x := vector(n) :
xs := 0 :
Er := E :
Ert : vector(tmax) :
for i from 1 to n do
    D[i, 1] := d[i] :
    x[i] := 0 :
od:
for t from 1 to tmax do
    X[1, t] := D[1, t]/n :
    x[1] := x[1] + X[1, t] :
    Er := Er−X[1, t] :
    for i from 2 to n do
        X[i, t] := X[i-1, t] + (D[i, t]-D[i-1, t])/(n-i + 1) :
        x[i] := x[i] + X[i, t] :
        Er := Er−X[i, t] :
    od:
    for i from 1 to n do
        D[i, t + 1] := min(D[i, t]-X[i, t], Er) :
    od:
    Ert[t] := Er :
    if Ert[t] = Ert[t-1] then break fi:
    if Er ≤ 0 then break fi:
od:
T := t :
end proc:
```

Figure 3: Subroutine GIEVr

```
GiEVMC := proc(N, G, MC)
AT; array(1 .. N, 1 .. G) :
 for n from 2 to N do
     for g from 1 to G do
          Ts := 0 :
          for mc from 1 to MC do
               GiEVr(1000, n, 10^(-g)) :
               Ts := Ts + T :
               aT := evalf(Ts/MC) :
               #print(T, Ts, `Mean T = `, aT)
          od:
          AT[n, g] := aT :
          print(`n = `, n, `; gap = `, g, `; Average T = `, aT) :
     od:
od:
 end proc:
```

Figure 4: Procedure GIEVMC