授業を補助するプログラム (4)
—破 産 問 題—

社会情報学科　行 方 常 幸

1．はじめに

私が担当する数理的な科目（例えば「意思決定論」等）においては，その内容を理解するために，多くの数値例を自分で解くことが必要である。しかしながら，人間の手計算で解ける問題は，小規模なものに限られる。また，手計算で解ける問題でも，自分で求めた答えが正解であるかをチェックすることも容易ではない。正解が不明なため，演習を途中であきらめた経験を持つのは私だけではないであろう。これに対処するために「破産問題」の解を計算するプログラムを作成したので本稿で紹介する。中規模程度以下の問題のデータを入力し，該当するタブを表示させれば，種々の解を計算するプログラムである。解法の違いを視覚で捉えることができるように工夫した。このプログラムを有効に利用することにより，計算の手間にとらわれず，種々の解の定性的な違いに注意の焦点を集めることが可能となる。

2．「破産問題」とその解

この節では「破産問題」とその解を説明する。

表1　各プレイヤーの要求額

プレイヤー	1	2	3	4
要 求 額	100	300	400	500

次のような例を考える：ある人が資産1,000万円を残し死亡した。この人には4人（この人たちをプレイヤーと呼ぶ）に借金がありその額（要求額と呼び，単位は万円とする）は表1のように与えられている。資産が要求額の和よりも少ないので，各プレイヤーに資産をいかに分けるべきか？　という問題が発生する。これが破産問題である。

一般的に，プレイヤーの集合を $N:=\{1,\ldots,n\}$，また，$0<E\leq\sum_{j\in N}d_j$, $d_j>0$ $(\forall j\in N)$ とし，資産を E，要求額を $d:=(d_1,\ldots,d_n)$ とする時，破産問題を $(E;d)$ で表す。破産問題の解 f は破産問題 $(E;d)$ に E の配分 $f(E;d):=(f_1(E;d),\ldots,f_n(E;d))$ を対応させる関数である。E の配分であることから f は $\sum_{j\in N}f_j(E;d)=E$ また，問題の内容から

$$0\leq f_j(E;d)\leq d_j(\forall j\in N) \tag{1}$$

を満たす。

本稿で扱う解は，比例配分法，Head 法，Leveling 法，仁，シャープレイ値，τ–値の6つ[1]である。仁，シャープレイ値，τ–値は破産問題を部分集合として含むより広い譲渡可能効用を持つ提携形ゲームの解である。

比例配分法（Prop）：

E を d に比例して配分する方法である。これは E を配分する際に，各プレ

1）これらの解の詳細は Young，Moulin，Driessen を参照。

イヤーの要求額のどの１円も同等の貢献をしているという考えによっている。式で書くと

$$\mathrm{Prop}_j(E;d):=\frac{d_j}{\sum_{i\in N} d_i}E\ (\forall j\in N)$$

Head 法（Head）：

各プレイヤーを同等に扱い，E をなるべくプレイヤーに均等に配分する方法である。均等にしたいが条件(1)を満たす必要があるので，次のように若干の修正が必要である。

$$\mathrm{Head}_j(E;d):=\min\{\lambda,d_j\}\ (\forall j\in N)$$

ただし，λ は $\sum_{j\in N}\mathrm{Head}_j(E;d)=E$ を満たす。

Leveling 法（Lev）：

Head 法と双対的に各プレイヤーの不足分 $d_j-\mathrm{Lev}_j(E;d)$ をなるべく均等に配分する方法である。式で書くと

$$\mathrm{Lev}_j(E;d):=d_j-\min\{\lambda,d_j\}\ (\forall j\in N)$$

ただし，λ は $\sum_{j\in N}\mathrm{Lev}_j(E;d)=E$ を満たす。

仁（Nuc）：

E が小さい時は Head 法の考え方で，E が大きい時は Leveling 法の考え方で配分する。式で書くと

$$\mathrm{Nuc}_j(E;d):=\begin{cases}\min\left\{\lambda,\dfrac{d_j}{2}\right\} & \text{for } E\leq\dfrac{1}{2}\sum_{i\in N}d_i\\[2ex] d_j-\min\left\{\lambda,\dfrac{d_j}{2}\right\} & \text{for } E\geq\dfrac{1}{2}\sum_{i\in N}d_i\end{cases}$$

ただし，λ は $\sum_{j\in N}\mathrm{Nuc}_j(E;d)=E$ を満たす[2)]。

2）詳しくは，補遺参照。

シャープレイ値（Sh）：

n 人のプレイヤーが順番に到着する。先着順に（まだ残っている資産があれば）自分の要求額（もし，要求額に満たなければ残っているもの全額）を受取る。$n!$ 通りの到着順における受取り額を平均したものがシャープレイ値である。式で書くと

$$\mathrm{Sh}_j(E;d):=\frac{1}{n!}\sum_{\pi\in\Pi}f_j^{\pi}(E;d)$$

$$f_j^{\pi}(E;d):=\min\left\{d_j,\ \max\left\{0,E-\sum_{i:\pi(i)<\pi(j)}d_i\right\}\right\}$$

ただし，Π は $n!$ 通りの順列の集合で，その要素 π において $\pi(i)$ はプレイヤー i が到着した順番を表す。

τ-値（Tau）：

比例配分に似た配分方法である。$E\leq\frac{1}{2}\sum_{i\in N}d_i$ の時：まず，各プレイヤーの要求額を E で抑える。すなわち，$d_j'=\min\{E,d_j\}$ $(\forall j\in N)$ とする。新たな破産問題 $(E;d')$ の比例配分を求める。その比例配分による配分が元の要求額の半分を超えるプレイヤーから，その超過分を取り，元の要求額の半分に抑える。元の要求額が最大であるプレイヤーは残りを貰う[3)]。式で書くと

$$\mathrm{Tau}_j(E;d):=\min\left\{\mathrm{Prop}_j(E;d'),\ \frac{1}{2}d_j\right\}\ (\forall j\in N)$$

$$\mathrm{Tau}_K(E;d):=E-\sum_{j\neq K}\mathrm{Tau}_j(E;d)$$

ただし，$d_K=\max\{d_j\mid j\in N\}$ である。 $E\geq\frac{1}{2}\sum_{i\in N}d_i$ の時：

$$\mathrm{Tau}_j(E;d)=d_j-\mathrm{Tau}_j\left(\sum_{i\in N}d_i-E;d\right)\ (\forall j\in N)$$

3）詳しくは，補遺を参照。

以上が，本稿で取り上げる6つの配分方法である。これらの中で，比例配分法，Head 法，Leveling 法，仁の4つの配分方法は優先法（priority method）としても解釈できる。優先法とは，各プレイヤー毎に，自分の要求額と現在自分に配分されている額のみに依存して決まる優先度が与えられていて，追加の資産をその時の優先度が一番高いところに配分する，という方法である。各々の優先度 $r(d,x)$ は表2に与えられている。d がプレイヤーの要求額で，x が現在配分されている額である。

表2　優　先　度

配分方法	優先度 $r(d,x)$
比例配分法	$\frac{d-x}{d}$
Head 法	$-x$
Leveling 法	$d-x$
仁	$\begin{cases} -x & \text{for } 0 \leq x < \frac{d}{2} \\ d-x & \text{for } \frac{d}{2} \leq x \leq d \end{cases}$

次節で紹介するプログラムでは比例配分法，Head 法，Leveling 法，仁の4つの配分方法が優先法であることを視覚的に理解できるように，図的解法を行っている。

3.「破産問題」プログラム

この節では「破産問題」を解くプログラムを紹介する。前節で導入した例を解くことにより，プログラムの利用法を説明し，プログラミングの時に特に留意した点を述べる。

このプログラムのアプリケーション版を実行したのが図1である。「データの入力」タブと「解の計算」タブがあるウィンドウが表示される。この「データの入力」タブでデータを入力する。見て明らかなように，必要ならばプレイヤーの人数を変え「変更」ボタンを押す。次に，資産 E と要求額 d を入力する。

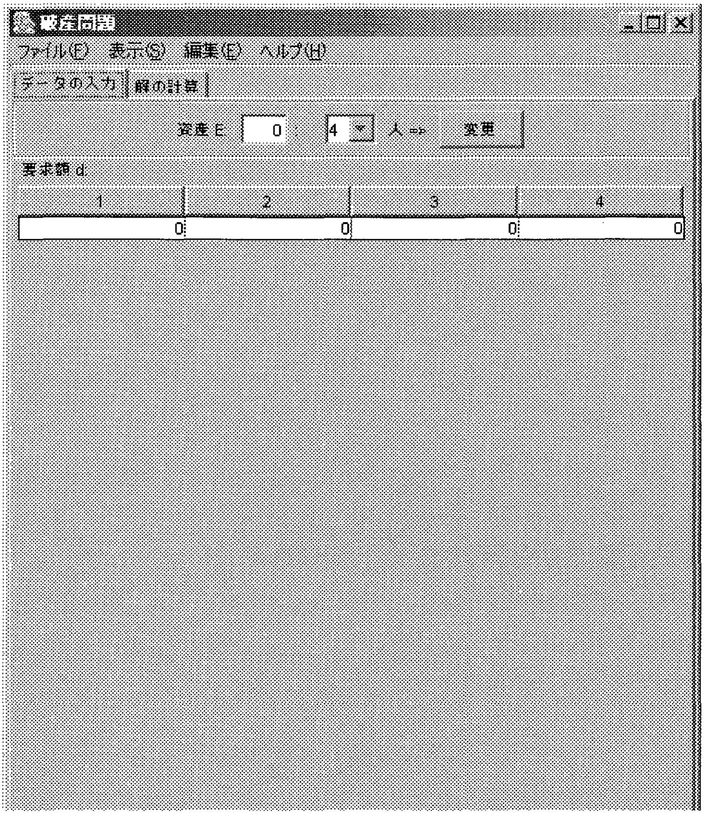


図1　「破産問題」起動時のウィンドウ

前節の数値例のデータを代入すると図2のようになる。「解の計算」タブを押すと解が図で表示される。この「解の計算」タブは「比例配分」から「τ–値」と「まとめ」の合計7つのタブからなる。

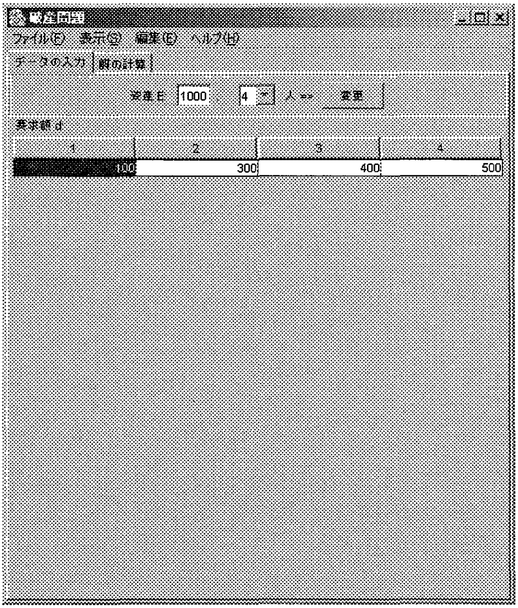
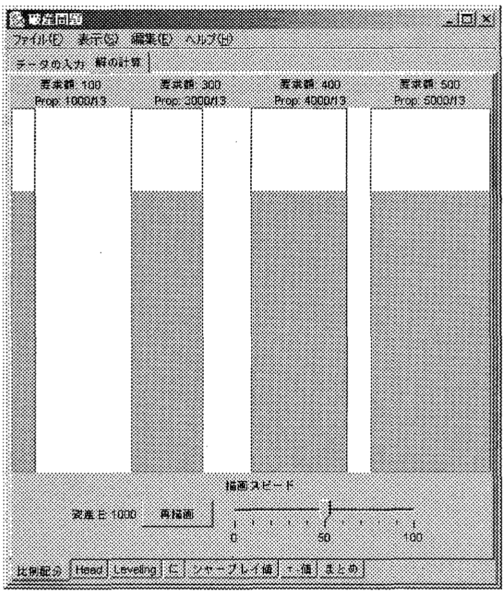
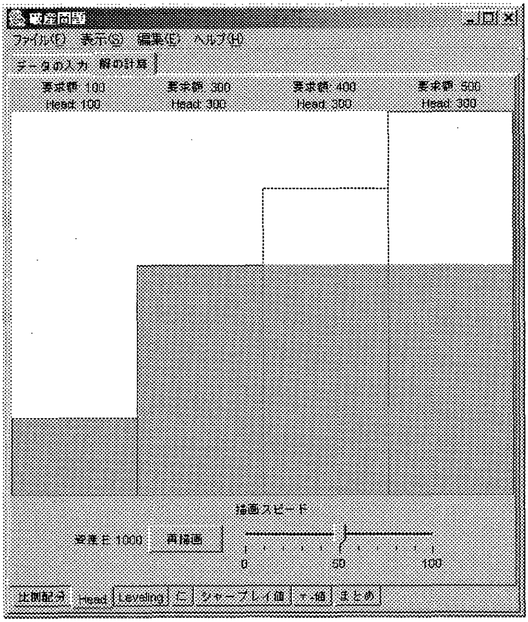
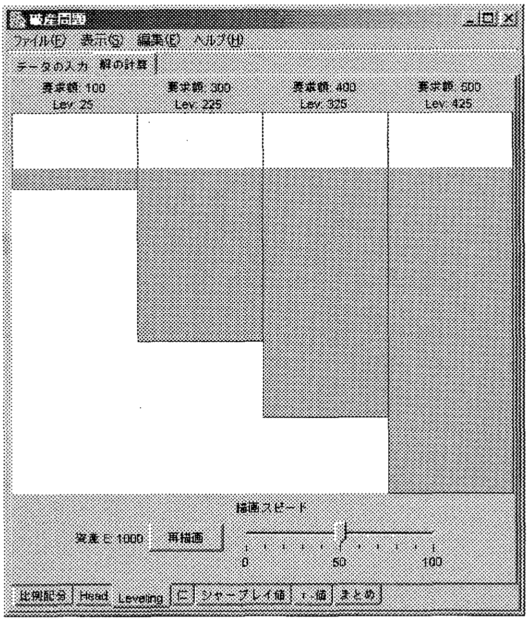
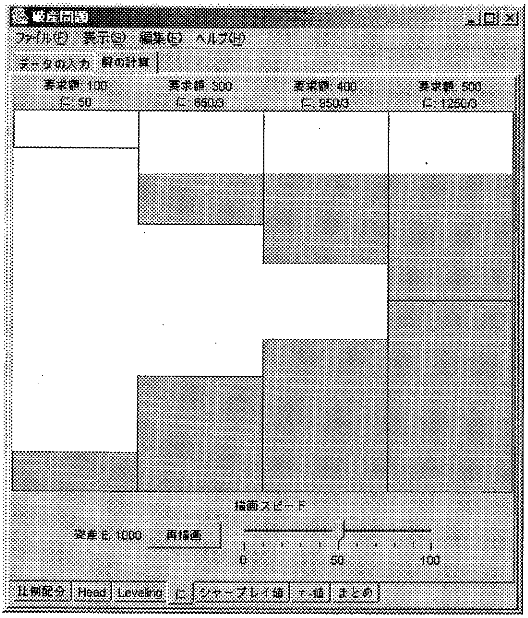

破産問題
ファイル(F)
表示(S)
編集(E)
ヘルプ(H)
データの入力
解の計算
資産 E
1000
4
人 =>
変更
要求額 d
1
2
3
4
100
300
400
500

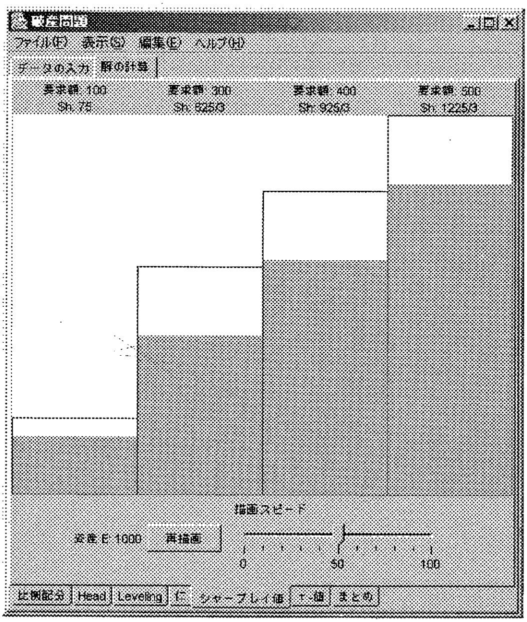
図2　データの入力

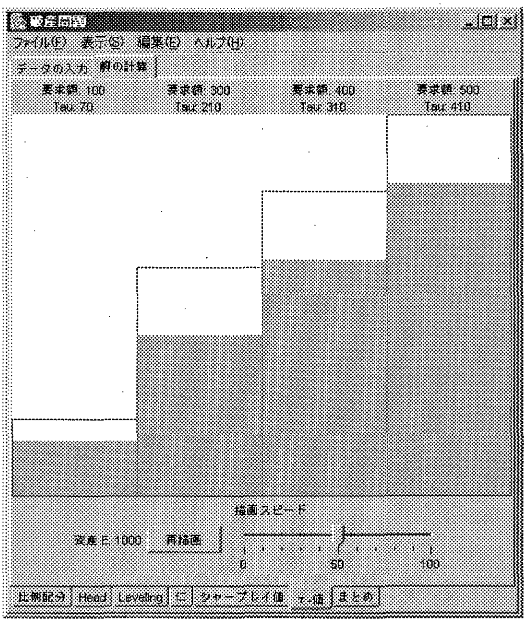

破産問題
ファイル(F)
表示(S)
編集(E)
ヘルプ(H)
データの入力
解の計算
要求額 100
Prop: 1000/13
要求額 300
Prop: 3000/13
要求額 400
Prop: 4000/13
要求額 500
Prop: 5000/13
描画スピード
資産 E 1000
再描画
0
50
100
比例配分
Head
Leveling
仁
シャープレイ値
τ-値
まとめ

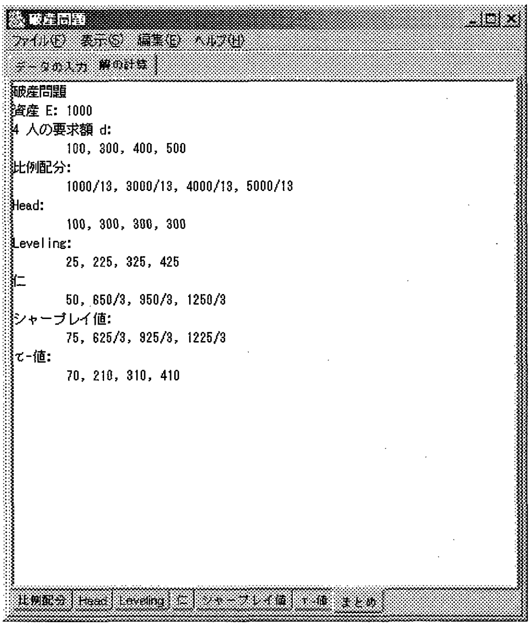

破産問題
ファイル(F) 表示(S) 編集(E) ヘルプ(H)
データの入力 解の計算
要求額: 100 Head: 100
要求額: 300 Head: 300
要求額: 400 Head: 300
要求額: 500 Head: 300
描画スピード
資産 E 1000
再描画
0 50 100
比例配分 Head Leveling C シャープレイ値 τ-値 まとめ


破産問題
ファイル(F) 表示(S) 編集(E) ヘルプ(H)
データの入力 解の計算
要求額: 100 Lev: 25
要求額: 300 Lev: 225
要求額: 400 Lev: 325
要求額: 500 Lev: 425
描画スピード
資産 E 1000
再描画
0 50 100
比例配分 Head Leveling C シャープレイ値 τ-値 まとめ


破産問題
ファイル(F) 表示(S) 編集(E) ヘルプ(H)
データの入力 解の計算
要求額: 100
仁: 50
要求額: 300
仁: 650/3
要求額: 400
仁: 950/3
要求額: 500
仁: 1250/3
描画スピード
資産 E: 1000
再描画
0
50
100
比例配分
Head
Leveling
仁
シャープレイ値
τ-値
まとめ

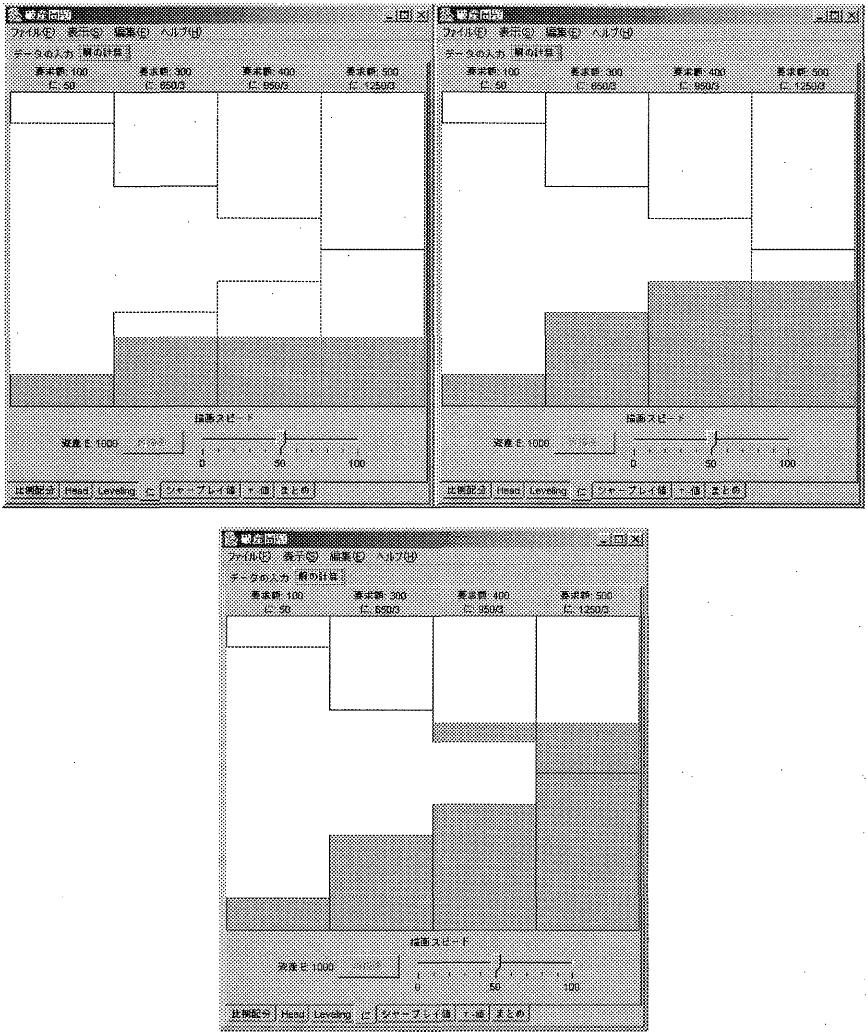

破産問題
ファイル(F) 表示(S) 編集(E) ヘルプ(H)
データの入力 解の計算
要求額: 100
Sh: 75
要求額: 300
Sh: 625/3
要求額: 400
Sh: 925/3
要求額: 500
Sh: 1225/3
描画スピード
資産 E: 1000
再描画
0
50
100
比例配分
Head
Leveling
仁
シャープレイ値
τ-値
まとめ


破産問題
ファイル(F) 表示(S) 編集(E) ヘルプ(H)
データの入力 解の計算
要求額 100
Tau 70
要求額 300
Tau 210
要求額 400
Tau 310
要求額 500
Tau 410
描画スピード
資産 E 1000
再描画
0
50
100
比例配分 Head Leveling 仁 シャープレイ値 τ-値 まとめ


破産問題
ファイル(F) 表示(S) 編集(E) ヘルプ(H)
データの入力 解の計算
破産問題
資産 E: 1000
4 人の要求額 d:
100, 300, 400, 500
比例配分:
1000/13, 3000/13, 4000/13, 5000/13
Head:
100, 300, 300, 300
Leveling:
25, 225, 325, 425
仁
50, 650/3, 950/3, 1250/3
シャープレイ値:
75, 625/3, 925/3, 1225/3
τ-値:
70, 210, 310, 410
比例配分 Head Leveling 仁 シャープレイ値 τ-値 まとめ

各解のタブにはその解による配分額の値とその図的表現が表示され，「まとめ」タブにはすべての解による配分額がまとめられている。

「比例配分」から「仁」までの４つのタブでは下部にある「再描画」ボタン



途中の３時点の状態

が利用できる。前節で述べたように，これらの配分方法は優先法とも解釈できるので，その解釈法で図示させるボタンである。すなわち，追加の資産をその時点で（自分の要求額と現在の配分額のみで決定される）優先度が高いプレイヤーに配分する，というように図示できる。「仁」タブで（必要ならば「描画スピード」をスライダーで調節し）「再描画」ボタンを押すと（「再描画」ボタンが利用不可能，すなわち，淡色表示になり），水色の部分が下方から上方へと移動し最終の配分額に達する。これらの図において，各棒の水色の部分の上端の縦軸方向の座標が現在の優先度（下に行くほど高い）になるように図は描かれている。優先法に従って各棒の水色の部分の上端が（可能な限り）水平になるように上へ移動していく。

このような図を描く際の注意点と時間の経過に従って表示を変化させる（一定時間毎に処理をさせる）方法について以下に述べる。

図を描く際の注意点：

「仁」等のタブの背景が白い部分（JPanel）に図（赤色の棒の境界と水色の棒）を描いている。同じタブ上にある，例えば，「要求額：100」という文字列は描く必要がある時に，一度描けば，（他のウィンドウの裏になりその後また

図を描くコード

```
private JPanel jPanelFigure = new JPanel ( ) {
    public void paintComponent (Graphics g) {
    super.paintComponent (g) ;
    mypaintComponent (g) ;
    }
};
private void mypaintComponent (Graphics g) {
    /*  gに現時点でのデータを利用して（赤と水色の部分の）棒を描く */
}
```

表に現れた時に）自動的に再描画される。しかしながら，図は以下に述べるようにしないと，再描画されない。

図を描く JPanel を作る時にメソッド paintComponent を定義し，その中で実際に描画を行うメソッド（ここでは mypaintComponent という名前にした）を呼び出す。mypaintComponent の中で，呼び出された時点での図を実際に描く（「図を描くコード」参照）。

一定時間毎に処理をさせる方法：

「再描画」ボタンが押されたら，時間の経過に従って水色の部分が変化する。これは次のように実現している。「再描画」ボタンが押されたら，（処理中に再度同じ処理を依頼されることを想定していないため）このボタンを利用不可能にし，一定時間毎に通知するタイマーを起動させる（『「再描画」ボタンが押されたら』を参照）。

「再描画」ボタンが押されたら

```
jButtonAnimate.setEnabled (false) ;
  //「再描画」ボタンを利用不可能にする
timer = new  Timer ( ) ;
  // タイマーを作り
timer.scheduleAtFixedRate (new MyTimerTask ( ), 0, 100) ;
  // 100/1000秒毎にMyTimerTaskのrunメソッドを呼び出すように設定
```

このタイマーの通知により実行される run メソッドの中で，水色の棒の長さを計算し，その結果を描画させる。具体的には，描画の繰返し中ならば，その時点での仁を計算し，結果を実際に表示させるために，jPanelFigure に更新，再描画するように命令する（これにより，上述の mypaintComponent が実行される）。描画の繰返しが終了したならば，タイマーをキャンセルし，「再描画」ボタンを利用可能にする（「run メソッド」を参照）。

runメソッド

```
private class MyTimerTask extends
  TimerTask {
    public void run ( )  {
      if (/* 描画の繰返し中か */) {
        // 現時点の仁を計算
        jPanelFigure.revalidate ( ) ;
        jPanelFigure.repaint ( ) ;
      } else {
        timer.cancel ( ) ;
        // タイマーをキャンセル
        jButtonAnimate.setEnabled (true) ;
        //「再描画」ボタンを利用可能にする
      }
    }
}
```

4．補　遺

この補遺において破産問題の仁と τ–値を求める。破産問題 $(E;d)$ から破産ゲーム $(N,v_{E;d})$ が次のように定義される。

$$v_{E;d}(S):=\begin{cases}\max\{E-d(N-S),0\} & \text{for } \forall S\subset N, S\neq\emptyset\\ 0 & \text{for } S=\emptyset\end{cases}$$

ただし，$d(S):=\begin{cases}\sum_{j\in S} d_j & \text{for } \forall S\subset N, S\neq\emptyset\\ 0 & \text{for } S=\emptyset\end{cases}$ である。また，便宜のため

$$f(j):=v_{E;d}(\{j\})=\max\{0,E-d(N)+d_j\}\ (\forall j\in N)$$

$$\Delta(j):=v_{E;d}(N)-v_{E;d}(N-\{j\})=\min\{E,d_j\}\ (\forall j\in N)$$

とおく。また，$\{d_j\,|\,j\in N\}$ の最大値を与える j を K とおく[4)]。

(i)　$E-d(N)+d_K \leq 0$ の場合：

K の決め方より $f(j)=0\,(\forall j \in N)$ となる。

(ii)　$E-d(N)+d_K>0$ の場合：

$f(K)=E-d(N)+d_K>0$である。また，$\sum_{j \neq K} d_j<E$ より $d_j \leq E\,(\forall j \neq K)$ となり，従って $\Delta(j)=d_j\,(\forall j \neq K)$ である。さらに $E \leq \frac{d(N)}{2}$ ならば $d(N)-E \geq E \geq d_j$ $(\forall j \neq K)$，$E \leq \frac{d(N)}{2} \leq d(N)-E<d_K$ となる。従って，$f(j)=0\,(\forall j \neq K)$，$\Delta(K)=E$ である。

仁の導出：

概略を述べる。破産ゲーム $(N, v_{E;d})$ の仁は次のように定義される。まず，$\sum_{j \in N} x_j=E$ である $x=(x_1,\ldots,x_n)$ に対して，不満 $e(S,x):=v_{E;d}(S)-\sum_{j \in S} x_j$ を定義する。$\{e(S,x) \mid \forall S \subset N\}$ の要素を大きいものから順に並べたものを $\theta(x)$ とおく。$\left\{x \middle| f(j) \leq x_j \leq \Delta(j)\,(\forall j \in N), \sum_{j \in N} x_j=E\right\}$ の領域において辞書式順序[5]の意味で $\theta(x)$ を最小にする x が仁である。

まず，$v_{E;d}(S) \leq \max\left\{\sum_{j \in S} f(j), E-\sum_{j \in N-S} \Delta(j)\right\}$ が成立する[6]。従って，

$$v_{E;d}(S)-\sum_{j \in S} x_j \leq \max\left\{\sum_{j \in S}(f(j)-x_j), \sum_{j \in N-S}(x_j-\Delta(j))\right\}$$

$v_{E;d}(\{j\})-x_j=f(j)-x_j$ と $v_{E;d}(\mathrm{N}-\{j\})-\sum_{i \in N-\{j\}} x_i=x_j-\Delta(j)$ に注意すると

$$\theta(x)=(\max\{\max\{f(j)-x_j, x_j-\Delta(j)\} \mid j \in N\},\ldots)$$

すなわち，$\theta(x)$ の最初の n 個の要素は $\{\min\{x_j-f(j), \Delta(j)-x_j\} \mid j \in N\}$ の要

4）最大値を与える j が 2 個以上ある場合は，適当に 1 つを選ぶ。

5） 2 つのベクトル $\theta(x)$ と $\theta(y)$ の要素を左から順番に比較し，異なる最初の要素の大小を $\theta(x)$ と $\theta(y)$ の大小とする。

6）例えば，Namekata *et al.* 参照。

素を小さい順に並べ符号を変えたものである。従って，まず，これらをなるべく大きくする。$0\leq\lambda_j\leq\frac{\Delta(j)-f(j)}{2}$ である λ_j に対して，x_j の方程式

$$\min\{x_j-f(j),\Delta(j)-x_j\}=\lambda_j$$

を解くと，

$x_j(\lambda_j)=f(j)+\lambda_j\left(\leq\frac{f(j)+\Delta(j)}{2}\right)$ または $x_j(\lambda_j)=\Delta(j)-\lambda_j\left(\geq\frac{f(j)+\Delta(j)}{2}\right)$

となる。これらは $\lambda\geq0$ を利用して，$x_j(\lambda)=\min\left\{f(j)+\lambda,\frac{f(j)+\Delta(j)}{2}\right\}$ または $x_j(\lambda)=\max\left\{\Delta(j)-\lambda,\frac{f(j)+\Delta(j)}{2}\right\}$ と書き直すことができる。$\{\lambda_j|j\in N\}$ をなるべく大きくし，$\sum_{j\in N}x_j(\lambda)=E$ とするには

$$x_j(\lambda)=\begin{cases}\min\left\{f(j)+\lambda,\frac{f(j)+\Delta(j)}{2}\right\} & \text{for } E\leq\frac{\sum_{i\in N}(f(i)+\Delta(i))}{2}\\ \max\left\{\Delta(j)-\lambda,\frac{f(j)+\Delta(j)}{2}\right\} & \text{for } E\geq\frac{\sum_{i\in N}(f(i)+\Delta(i))}{2}\end{cases}\tag{2}$$

ただし，λ は $\sum_{j\in N}x_j(\lambda)=E$ を満たす。この $(x_j(\lambda))_{j\in N}$ が2節『「破産問題」とその解』の Nuc であることに示せば，これが $\theta(x)$ を辞書的順序の意味で最小にする x，すなわち，仁であることが分かる。

(i)　$E-d(N)+d_K\leq0$ の場合：

$N_1:=\{j\in N|d_j\leq E\},N_2:=N-N_1$ とおくと，

$$\frac{d(N)}{2}\begin{cases}\geq E & \text{for } |N_2|\geq2\\ \geq\frac{d_K+E}{2}>\mathrm{E} & \text{for } N_2=\{K\}\end{cases}$$

$f(j)=0,\Delta(j)=\min\{E,d_j\}\,(\forall j\in N)$ であるので，

$$\frac{\sum_{j\in N}(f(j)+\Delta(j))}{2}$$

$$=\frac{\sum_{j\in N_1}d_j+|N_2|E}{2}\begin{cases}\geq E & \text{for } |N_2|\geq 2\\ =\dfrac{d(N)-d_K+E}{2}\geq E & \text{for } N_2=\{K\}\\ =\dfrac{d(N)}{2} & \text{for } N_2=\emptyset\end{cases}$$

となる。従って，$N_2\neq\emptyset$ の時，$E\leq\dfrac{d(N)}{2}$ であり，

$$x_j(\lambda)=\begin{cases}\min\left\{\lambda,\dfrac{d_j}{2}\right\} & \text{for } j\in N_1\\ \min\left\{\lambda,\dfrac{E}{2}\right\} & \text{for } j\in N_2\end{cases}$$

$\sum_{j\in N}x_j\left(\dfrac{E}{2}\right)=\dfrac{d(N_1)}{2}+|N_2|E\geq E$ であるので，λ は $\dfrac{E}{2}$ を超えることはない。

従って，$\min\left\{\lambda,\dfrac{E}{2}\right\}=\min\left\{\lambda,\dfrac{d_j}{2}\right\}(j\in N_2)$ としても良い。

$N_2=\emptyset$ の時，次のように書ける。

$$x_j(\lambda)=\begin{cases}\min\left\{\lambda,\dfrac{d_j}{2}\right\} & \text{for } E\leq\dfrac{d(N)}{2}\\ \max\left\{d_j-\lambda,\dfrac{d_j}{2}\right\} & \text{for } E\geq\dfrac{d(N)}{2}\end{cases}$$

(ii) $E-d(N)+d_K>0$ の場合：

$M_1:=\{j\in N|E-d(N)+d_j\leq 0\}, M_2:=N-M_1$ とおく。$K\in M_2$ であり，

$$f(j)=\begin{cases}0 & \text{for } j\in M_1\\ E-d(N)+d_j & \text{for } j\in M_2\end{cases}$$

$$\Delta(j)=\begin{cases}d_j & \text{for } j\neq K\\ \min\{d_K,E\} & \text{for } j=K\end{cases}$$

である。(A) $E \leq \frac{d(N)}{2}$，(B) $E > \frac{d(N)}{2}, d_K < E$，(C) $E > \frac{d(N)}{2}, d_K \geq E$ の場合に分けて考察する。

(A)　$E \leq \frac{d(N)}{2}$ の時：

$M_2 = \{K\}, \Delta(K) = E$ であるので，$\frac{\sum_{j \in N}(f(j) + \Delta(j))}{2} = E$ となる。従って，

(2)の最初の表現を利用して，

$$x_j(\lambda) = \begin{cases} \min\left\{\lambda, \frac{d_j}{2}\right\} & \text{for } j \neq K \\ \min\left\{E - d(N) + d_K + \lambda, \frac{2E - d(N) + d_K}{2}\right\} & \text{for } j = K \end{cases}$$

ただし，$\lambda \geq \frac{d(N) - d_K}{2}$ である。

$E - \frac{d(N) - d_K}{2} > \frac{d(N) - d_K}{2} \geq \frac{d_j}{2} (\forall j \neq K)$ に注意すればこれは次の解と一致する。

$$x_j(\lambda) = \min\left\{\lambda, \frac{d_j}{2}\right\}$$

(ただし，$\lambda = E - \frac{d(N) - d_K}{2} = \frac{d_K}{2} - \left(\frac{d(N)}{2} - E\right) \leq \frac{d_K}{2}$)

(B)　$E > \frac{d(N)}{2}, d_K < E$ の時：

$\Delta(K) = d_K, d(M_2) \leq d(N), -d(N) \leq -E$ であるので，

$$\frac{\sum_{j\in N}(f(j)+\Delta(j))}{2}=\frac{d(N)+d(M_2)+(E-d(N))|M_2|}{2}$$

$$\begin{cases} =\dfrac{E+d_K}{2} & \text{for } M_2=\{K\} \\ \le\dfrac{|M_2|}{2}E-\dfrac{(|M_2|-2)}{2}d(N) & \text{for } |M_2|\ge 2 \end{cases}$$

$$\le E$$

(2)は

$$x_j(\lambda)=\begin{cases} \max\left\{d_j-\lambda,\dfrac{d_j}{2}\right\} & \text{for } j\in M_1 \\ \max\left\{d_j-\lambda,d_j-\dfrac{d(N)-E}{2}\right\} & \text{for } j\in M_2 \end{cases}$$

となる。まず，次の方程式の解を求める。

$$\sum_{j\in N}y_j(\lambda)=E$$

$$y_j(\lambda)=\max\left\{d_j-\lambda,\frac{d_j}{2}\right\}\left(=d_j-\min\left\{\lambda,\frac{d_j}{2}\right\}\right)$$

$d(N)\ge E>\dfrac{d(N)}{2}$, $\sum_{j\in N}y_j(0)=d(N)$, $\sum_{j\in N}y_j\left(\dfrac{d_K}{2}\right)=\dfrac{d(N)}{2}$ であり，$\sum_{j\in N}y_j(\lambda)$ は λ の単調減少関数であるので，$\sum_{j\in N}y_j(\lambda^*)=E$ となる $\lambda^*\left(<\dfrac{d_K}{2}\right)$ が一意に存在する。 $L:=\left\{j\in N\,\middle|\,d_j-\lambda^*>\dfrac{d_j}{2}\right\}$ とおけば， $K\in L$ であり，

$$\sum_{j\in L}(d_j-\lambda^*)+\sum_{j\in N-L}\frac{d_j}{2}=E$$

$$\lambda^*=\frac{d(N)+d(L)-2E}{2|L|}$$

が成り立つ。また，

$$\lambda^* \begin{cases} = \dfrac{d(N) + d_K - 2E}{2} & \text{for } L = \{K\} \\ \le \dfrac{d(N) - E}{|L|} & \text{for } |L| \ge 2 \end{cases}$$

$$\le \frac{d(N) - E}{2} < \frac{d_j}{2} \; (\forall j \in M_2)$$

以上から，$L \supset M_2$ であり，結局，$x(\lambda^*) = y(\lambda^*)$ であることが分かる。

(C)　$E > \dfrac{d(N)}{2}, d_K \ge E$ の時：

$\Delta(K) = E$ である。$d(N) - E \ge d(N) - d_K \ge d_j \; (\forall j \ne K)$ より $M_2 = \{K\}$ となる。

$\dfrac{\sum_{j \in N} (f(j) + \Delta(j))}{2} = E$ となるから，

(2)の2つ目の表現を利用して

$$x_j(\lambda) = \begin{cases} \max\left\{d_j - \lambda, \dfrac{d_j}{2}\right\} & \text{for } j \ne K \\ \max\left\{E - \lambda, \dfrac{2E - d(N) + d_K}{2}\right\} & \text{for } j = K \end{cases}$$

ただし，$\lambda \ge \dfrac{d(N) - d_K}{2}$ である。$\dfrac{d(N) - 2E}{2} + \dfrac{d_K}{2} \le \dfrac{d_K}{2}$ と $\dfrac{d(N) - 2E}{2} + \dfrac{d_K}{2}$

$= \dfrac{d(N) - E}{2} + \dfrac{d_K - E}{2} \ge \dfrac{d_j}{2} \; (\forall j \ne K)$ に注意すると，これは次の解と一致する。

$$x_j(\lambda) = \max\left\{d_j - \lambda, \frac{d_j}{2}\right\}$$

（ただし，$\lambda = \dfrac{d(N) - 2E + d_K}{2}$）

τ-値の導出：

この破産ゲームは semiconvex であり，semiconvex なゲームの τ-値

$\mathrm{Tau}\,(E;d) = (\mathrm{Tau}_1(E;d), \ldots, \mathrm{Tau}_n(E;d))$ は

$$\mathrm{Tau}_j(E;d):=f(j)+\frac{\Delta(j)-f(j)}{\sum_{i\in N}(\Delta(i)-f(i))}\left(E-\sum_{i\in N}f(i)\right)(\forall j\in N) \tag{3}$$

となる。$\mathrm{Tau}(\sum_{i\in N}d_i-E;d)=d-\mathrm{Tau}(E;d)$ が成立することが分かっているので，$0\leq E\leq\frac{\sum_{i\in N}d_i}{2}$ の範囲で $\mathrm{Tau}(E;d)$ を求めればよい。次の2つの場合に分けて考察する。(i) $E-d(N)+d_K\leq 0$ の場合と(ii) $E-d(N)+d_K>0$ の場合である。

(i) $E-d(N)+d_K\leq 0$ の場合：

$f(j)=0(\forall j\in N)$ であるので，

$$\mathrm{Tau}_j(E;d)=\frac{d_j'}{\sum_{i\in N}d_i'}E(\forall j\in N)$$

ただし，$d_j'=\min\{E,d_j\}(\forall j\in N)$ である。

次に，$\mathrm{Tau}_j(E;d)=\frac{d_j'}{\sum_{i\in N}d_i'}E\leq\frac{1}{2}d_j$ を示す。$\frac{d_j}{d_j'}\geq 1$ であることに注意すると，$E\leq\frac{\sum_{i\in N}d_i'}{2}$ を示せば十分である。$N_1:=\{j\in N|d_j\leq E\}$ とおくと

$$\frac{\sum_{i\in N}d_i'}{2}=\frac{\sum_{i\in N_1}d_i+(n-|N_1|)E}{2}\geq E$$

である。何故なら，$N_1=N$ の時は仮定 $E\leq\frac{d(N)}{2}$ より成立する。$n-|N_1|\geq 2$ の時は $d_i>0$ より成立する。$E-d(N)+d_K\leq 0$ より $\sum_{j\neq K}d_j+E\geq 2E$ であり，$N_1=N-\{K\}$ の時も成立する。

従って，$\mathrm{Tau}_j(E;d)=\mathrm{Prop}_j(E;d')=\min\left\{\mathrm{Prop}_j(E;d'),\frac{1}{2}d_j\right\}$ であることが示された。$\mathrm{Tau}_K(E;d)=E-\sum_{j\neq K}\mathrm{Tau}_j(E;d)$ は当然成立している。

(ii)　$E-d(N)+d_K>0$ の場合：

$$f(j)=\begin{cases} 0 & \text{for } j\neq K \\ E-d(N)+d_K & \text{for } j=K \end{cases}$$

$$\Delta(j)=\begin{cases} d_j & \text{for } j\neq K \\ E & \text{for } j=K \end{cases}$$

である。これらを(3)へ代入すると，

$$\mathrm{Tau}_j(E;d)=0+\frac{d_j}{2(d(N)-d_K)}(d(N)-d_K)=\frac{1}{2}d_j(\forall j\neq K)$$

$$\begin{aligned}\mathrm{Tau}_K(E;d)&=E-d(N)+d_K+\frac{d(N)-d_K}{2(d(N)-d_K)}(d(N)-d_K)\\&=E-\frac{d(N)-d_K}{2}\\&=E-\sum_{j\neq K}\mathrm{Tau}_j(E;d)\end{aligned}$$

また，$d'_j=\min\{E;d_j\}=\begin{cases} d_j & \text{for } j\neq K \\ E & \text{for } j=K \end{cases}$ に注意すると

$$\mathrm{Prop}_j(E;d')=\begin{cases} \dfrac{d_j}{E+d(N)-d_K}E & \text{for } j\neq K \\[2ex] \dfrac{E}{E+d(N)-d_K}E & \text{for } j=K \end{cases}$$

$\dfrac{E}{E+d(N)-d_K}>\dfrac{1}{2}$ であるので，

$$\mathrm{Tau}_j(E;d)=\min\left\{\mathrm{Prop}_j(E;d'),\frac{1}{2}d_i\right\}(\forall j\neq K)$$

$$\mathrm{Tau}_K(E;d)=E-\sum_{j\neq K}\mathrm{Tau}_j(E;d)$$

が成立していることが分かる。

5. おわりに

本稿では「破産問題」の6つの解，比例配分法，Head法，Leveling法，仁，シャープレイ値，τ-値を計算するプログラムを紹介した。特徴は，解を図で表示した点である。これにより，これら6つの解の考え方の違いが視覚を通じて理解できるようになり，適用すべき状況に依存して，どの解を利用すべきかの判断が容易になることが期待される。

参考文献

[1] Young, H.P.: Equity－In Theory and Practice, Princeton University Press, 1995.

[2] Moulin, H.: Axioms of Cooperative Decision Making, Cambridge University Press, 1988.

[3] Driessen, T.S.H.: Cooperative Games, Solutions and Applications, Kluwer Academic Publishers, 1988.

[4] Namekata, T. & T.S.H Driessen: Bargaining Property of Nucleolus and τ-Value in a Class of TU-Games, Computers & Mathematics with Applications **41**, 703-721, (2001).