Bingo Pricing: a Game Simulation and Evaluation using the Derivatives Approach

Antonio Annibali

Francesco Bellini

University of Rome 'La Sapienza'
Faculty of Economics
Department of Mathematics for Economic, Financial and Insurance Decisions
Via del Castro Laurenziano, 9 (00139) Roma / Italia
aannib@scec.eco.uniroma1.it
fbellini@scec.eco.uniroma1.it

Keywords: computation, nested arrays, game, probability, financial derivatives

JEL classification: Game Theory and Bargaining Theory (C7), Mathematical Methods and Programming (C6)

1. Introduction

The **Bingo** game is well known and played all over the world. Its main feature is the sequential drawing without repetition of a set of **numbers**. Each of these numbers is compared to the numbers contained in the boxes printed on the different **rows** (and **columns**) of the **score-cards** owned by the Bingo participants. The winner will be the participant that firstly is able to check all the boxes (numbers) into a row (**Line**) or into the entire score-card (**Bingo**).

Assuming that the score-card has a predetermined purchase price and that the jackpot is divided into two shares, respectively for the Bingo and the Line winner, it is evident that all the score-cards show the same starting value (**initial price**).

After each drawing, every score-card will have different values **(current price(s))** according with its probability to gain the Line and/or the Bingo. This probability depends from the number of checked boxes in the rows of the score-card and from the number of checked boxes in the rows of all the other playing score-cards.

The first aim of this paper is to provide the base data structure of the problem and to formalize the needed algorithms for the **initial price** and **current price** calculation. The procedure will evaluate the single score-card and/or the whole set of playing score-cards according to the results of the subsequent drawings. In fact, during the game development and after each drawing, it will be possible to know the value of each score-card in order to choose if maintain it or sell it out. The evaluation will work in accordance to the traditional Galilee's method of "*the interrupted game jackpot repartition*". This approach has been also mentioned by Blaise Pascal and Pierre de Fermat in their mail exchange about the "*jackpot problem*".

More advanced objective of the paper would be the application of the stock exchange techniques for the calculation of the **future price** of the score-card (and/or of a set of score-cards) that will have some checked numbers after a certain number of future drawings. In the same way will be calculated the value of the right to purchase or sell a score-card (and/or of a set of score-cards) at a pre-determined price (**option price**).

Especially during the prototyping phase, the modelling and the development of these kind of problems need the use of computational environments able to manage structured data and with high calculation skills.

The software that meet these requirements are APL, J™ and Matlab™, as for their capability to use *nested arrays* and for the endogenous parallelism features of the programming environments.

In this paper we will show the above mentioned issues through the use of **Apl2Win/IBM**Ô.

The formalisation of the game structure has been made in a general way, in order to foresee particular cases that act differently from the Bingo. In this way it is possible to simulate the traditional game with 90 numbers in the basket, 3 rows per 10 columns score-cards, 15 number for the Bingo and 5 numbers for the Line but already, for example, the Roulette with 37 (or 38) numbers, score-cards with 1 (or more) row and 1 column and Line with just 1 number.

2. Description of the game structure

In order to describe the game structure the following elements have been defined, in coherence with the variables used for the programming environment **Apl2Win**:

- **NUM** number of numbers contained in the **Bingo basket**
  (i.e: 90 in the traditional **Bingo** game, 37 or 38 in the **Roulette** game)

- **WUM** number of numbers **to be drawn**
  (i.e.: 90 in the **Bingo** game, 1 in the **Roulette**)

- **ROW** number of **rows** in a score-card
  (i.e.: 3 in the **Bingo** game, 1 or more than 1 in the **Roulette** game, considering a single number or a predefined set of numbers, such as, for example, even numbers or odd numbers)

- **COL** **range-width** vector of the single columns in a score-card
  (i.e.: 9, 10, 10, 10, 10, 10, 10, 10, 11 in the **Bingo** game, 1 in the **Roulette**)

  $$NUM = \sum_{h=1}^{[COL]} (COL)_h$$

  being **[COL]** the number of columns

- **LIN** number of checked box for the **Line win**
  (i.e.: 5 in the **Bingo** game, 1 in the **Roulette**)

- **BIN** number of checked box for the **Bingo win**
  (i.e.: 15 in the **Bingo** game)

  $$BIN = ROW \cdot LIN$$

- **SET** number of score-cards in a score-cards **subset**
  (i.e.: 6 in the **Bingo** game)

  $$SET = \frac{NUM}{BIN}$$

- **NCA** number of score cards participating to the game
- **NDR** vector of drawn number

$$(NDR)_h \qquad h = 1,2,...,[NDR] \leq WUM$$

being **[NDR]** the number of drawn numbers

- **CBB** vector containing the number of **checked box** for each score-card participating to the game

$$(CBB)_h \,|[NDR] \qquad h = 1,2,..., NCA$$

- **CBL** matrix containing the number of **checked box** in the different rows for each score-card participating to the game

$$(CBL)_{h,k} \,|[NDR] \qquad h = 1,2,\ldots, NCA; k = 1,2,\ldots, ROW$$

being

$$(CBB)_h \,|[NDR] = \sum_{k=1}^{ROW} ((CBL)_{h,k} \,|[NDR]) \qquad h = 1,2,..., NCA$$

- **NCB** vector containing the number of playing score-cards, having a determined number of **checked box**

$$(NCB)_{NN} \,|[NDR] = \sum_{h=1}^{NCA} ((CBB)_h \,|[NDR] = NN) \qquad NN = 0,1,...,BIN$$

being

$$\sum_{NN=0}^{BIN} ((NCB)_{NN} \,|[NDR]) = NCA$$

- **NCL** vector containing the number of rows in the playing score-cards, having a determined number of **checked box**

$$(NCL)_{NN} \,|[NDR] = \sum_{h=1}^{NCA} \sum_{k=1}^{ROW} ((CBL)_{h,k} \,|[NDR] = NN) \qquad NN = 0,1,...,LIN$$

being

$$\sum_{NN=0}^{LIN} (NCL)_{NN} \,|[NDR] = NCA \cdot ROW$$

- **PRC** **purchase price** of a single score-card
- **MNT** amount retained from the **game organiser**

$$MNT = NCA \cdot PRC$$

- **ALPHA** % of the gross jackpot for the **Bingo winner**
- **BETA** % of the gross jackpot for the **Line winner**

- **MNB** share of the jackpot for the **Bingo winner**

  $MNB = MNT \cdot ALPHA$

- **MNL** share of the jackpot for the **Line winner**

  $MNL = MNT \cdot BETA$

- **MNZ** share of the jackpot for the **winners**

  $MNZ = MNT \cdot (ALPHA + BETA)$

- **PRI** **initial price** of a single playing score-card

  $PRI = PRC \cdot (ALPHA + BETA)$

3. Performing and evaluating the game

Considering the definition of generalised binomial coefficient

<1> $$\boxed{\binom{m}{n} = \frac{\prod_{j=1}^{n}(m-n+j)}{\prod_{j=1}^{n} j} \qquad m,n \in \mathbb{N}} \quad \text{being} \quad \binom{m}{n} = 0 \quad \text{if} \quad (m<n),$$

the probability of the **1**-nth score-card (sole competitor with $b_1(z) = (CBB)_1 \,|\, [NDR] < b$ checked numbers after the drawing of $z = [NDR]$ numbers) to obtain the Bingo within the t-nth drawing results

<2> $$\boxed{p_{b_1}^{b}(z,t) = \frac{\binom{w-z-(b-b_1(z))}{t-z-(b-b_1(z))}}{\binom{w-z}{t-z}} = \frac{\binom{t-z}{b-b_1(z)}}{\binom{w-z}{b-b_1(z)}}} \qquad t = z+1, z+2, \ldots, w$$

having indicated with $w = WUM$ and $b = BIN$ and being

$$p_{b_1}^{b}(z,t) = 0 \qquad t = z+1, z+2, \ldots, z+b-b_1(z)-1$$

and the probability of the above mentioned score-card to obtain the Bingo at the t-nth drawing results

<3> $$\boxed{p_{b_1}^{b}(z,t) = p_{b_1}^{b}(z,t) - p_{b_1}^{b}(z,t-1) = \frac{\binom{t-z}{b-b_1(z)} - \binom{t-z-1}{b-b_1(z)}}{\binom{w-z}{b-b_1(z)}} = \frac{\binom{t-z-1}{b-b_1(z)-1}}{\binom{w-z}{b-b_1(z)}}} \qquad t = z+1, z+2, \ldots, w$$

being

$$p^b_{b_1}(z,t) = 0 \qquad t = z+1, z+2, \ldots, z+b-b_1(z)-1$$

$$p^b_{b_1}(z,t) = \sum_{h=z+1}^{t} p^b_{b_1}(z,h) \qquad t = z+1, z+2, \ldots, w$$

and consequently the victory probability of the sole competitor is (obviously)

<4> $$\boxed{\sum_{t=z+b-b_1(z)}^{w} p^b_{b_1}(z,t) = \frac{\sum\limits_{t=z+b-b_1(z)}^{w} \binom{t-z-1}{b-b_1(z)-1}}{\binom{w-z}{b-b_1(z)}} = \frac{\binom{w-z}{b-b_1(z)}}{\binom{w-z}{b-b_1(z)}} = w^b_{b_1}(z,w) = 1}$$

Given a couple of score-cards (the **1**-nth (playing with $b_1(z) < b$ checked number) and the **2**-nth score-card (with $b_2(z) < b$ checked numbers)), the probability of the **1**-nth score-card to obtain the Bingo at the t-nth drawing results

<5> $$\boxed{p^b_{b_1|b_1,b_2}(z,t) = \begin{cases} 0 \\ p^b_{b_1}(z,t) \cdot (1 - p^b_{b_2}(z,t)) \end{cases}} \qquad \begin{array}{c} t = z+1, z+2, \ldots, z+b-b_1(z)-1 \\ t = z+b-b_1(z), z+b-b_1(z)+1, \ldots, w \end{array}$$

and similarly, the probability of the **2**-nth score-card to obtain the Bingo at the t-nth drawing results

<6> $$\boxed{p^b_{b_2|b_1,b_2}(z,t) = \begin{cases} 0 \\ p^b_{b_2}(z,t) \cdot (1 - p^b_{b_1}(z,t)) \end{cases}} \qquad \begin{array}{c} t = z+1, z+2, \ldots, z+b-b_2(z)-1 \\ t = z+b-b_2(z), z+b-b_2(z)+1, \ldots, w \end{array}$$

and, finally, the probability of both the score-card to obtain the Bingo at the t-nth drawing results

<7> $$\boxed{p^b_{b_2 \wedge b_2 | b_1, b_2}(z,t) = \begin{cases} 0 \\ p^b_{b_1}(z,t) \cdot p^b_{b_2}(z,t) \end{cases}} \qquad \begin{array}{c} t = z+1, z+2, \ldots, z+b-\min(b_1(z)|b_2(z))-1 \\ t = z+b-\min(b_1(z)|b_2(z)), \ldots, w \end{array}$$

Distributing equally the probability of a simultaneous Bingo of both the score cards, the probability of the **1**-nth score-card to obtain the Bingo at the t-nth drawing results

<8> $$\boxed{p^b_{b_1|b_1,b_2}(z,t) = \begin{cases} 0 \\ p^b_{b_1}(z,t) \cdot (1 - p^b_{b_2}(z,t) + \dfrac{p^b_{b_2}(z,t)}{2}) \end{cases}} \qquad \begin{array}{c} t = z+1, z+2, \ldots, z+b-b_1(z)-1 \\ t = z+b-b_1(z), z+b-b_1(z)+1, \ldots, w \end{array}$$

and similarly, the probability of the **2**-nth score-card to obtain the Bingo at the t-nth drawing results

<9> $$\boxed{p^b_{b_2|b_1,b_2}(z,t) = \begin{cases} 0 \\ p^b_{b_2}(z,t) \cdot (1 - p^b_{b_1}(z,t) + \dfrac{p^b_{b_1}(z,t)}{2}) \end{cases}} \qquad \begin{array}{c} t = z+1, z+2, \ldots, z+b-b_2(z)-1 \\ t = z+b-b_2(z), z+b-b_2(z)+1, \ldots, w \end{array}$$

and consequently the probability to win the Bingo for the **1**-nth score-card is

<10> $$\boxed{\sum_{t=z+b-b_1(z)}^{w} p^b_{b_1|b_1,b_2}(z,t) = \sum_{t=z+b-b_1(z)}^{w} p^b_{b_1}(z,t) \cdot (1 - p^b_{b_2}(z,t) + \frac{p^b_{b_2}(z,t)}{2}) = p^b_{b_1|b_1,b_2}(z,w)}$$

and similarly the probability to win the Bingo for the **2**-nth score-card is

<11> $$\boxed{\sum_{t=z+b-b_2(z)}^{w} p^b_{b_2|b_1,b_2}(z,t) = \sum_{t=z+b-b_2(z)}^{w} p^b_{b_2}(z,t)\cdot\left(1-p^b_{b_1}(z,t)+\frac{p^b_{b_1}(z,t)}{2}\right) = p^b_{b_2|b_1,b_2}(z,w)}$$

obviously resulting

$$p^b_{b_1|b_1,b_2}(z,w)+p^b_{b_2|b_1,b_2}(z,w)=1$$

Given a set of N=NCA score-cards (in which the **h**-nth plays with $b_h(z)<b$ checked numbers), the probability of this score cards the obtain the Bingo at the t-nth drawing results

<12> $$p^b_{b_h|b_1,b_2,...b_N}(z,t)=\begin{cases}0 & t=z+1,z+2,...,z+b-b_h(z)-1\\ p^b_{b_h}(z,t)\displaystyle\prod_{j=1,j\neq h}^{N}(1-p^b_{b_j}(z,t)) & t=z+b-b_h(z),z+b-b_h(z)+1,...,w\end{cases}$$

the probability of this score cards-the obtain the Bingo at the t-nth drawing, simultaneously with another playing score-card, results

<13> $$p^b_{b_h\wedge b_{h_2}|b_1,b_2,...b_N}(z,t)=\begin{cases}0 & t=z+1,z+2,...,z+b-\min(b_h(z)|b_{h_2}(z))-1\\ p^b_{b_h}(z,t)\cdot p^b_{b_{h_2}}(z,t)\displaystyle\prod_{j=1,j\neq h,j\neq h_2}^{N}(1-p^b_{b_j}(z,t)) & t=z+b-\min(b_h(z)|b_{h_2}(z)),...,w\end{cases}$$

the probability of the above mentioned score-cards the obtain the Bingo at the t-nth drawing, simultaneously with two other playing score-cards, results

<14> $$p^b_{b_h\wedge b_{h_2}\wedge b_{h_3}|b_1,b_2,...b_N}(z,t)=\begin{cases}0 & t=z+1,z+2,...,z+b-\min(b_h(z)|b_{h_j}(z);j=2,3)-1\\ p^b_{b_h}(z,t)\cdot\displaystyle\prod_{j=2}^{3}p^b_{b_{h_j}}(z,t)\prod_{j=1,j\neq h,j\neq h_2,j\neq h_3}^{N}(1-p^b_{b_j}(z,t)) & t=z+b-\min(b_h(z)|b_{h_j}(z);j=2,3),...,w\end{cases}$$

and, finally, the probability of the entire set of score-cards to obtain the Bingo simultaneously at the t-nth drawing, results

<15> $$p^b_{b_1\wedge b_2\wedge...\wedge b_N|b_1,b_2,...b_N}(z,t)=\begin{cases}0 & t=z+1,z+2,...,z+b-\min(b_j(z);j=1,2,...,N)-1\\ \displaystyle\prod_{j=1}^{N}p^b_{b_j}(z,t) & t=z+b-\min(b_j(z);j=1,2,...,N),...,w\end{cases}$$

Distributing equally the probability of a simultaneous Bingo for every score-cards, the probability that the **h**-nth score-card to obtain the Bingo at the t-nth drawing, results

<16> $$\boxed{p^{b}_{b_h|b_1,b_2,\ldots b_N}(z,t)=\begin{cases}0\\ p^{b}_{b_h}(z,t)\prod_{j=1,\,j\neq h}^{N}(1-p^{b}_{b_j}(z,t))+\sum_{k=2}^{N}\sum_{h_2,h_3,\ldots,h_k}^{\binom{N-1}{k-1}}\dfrac{p^{b}_{b_h\wedge b_{h_2}\wedge\ldots\wedge b_{h_k}|b_1,b_2,\ldots b_N}(z,t)}{k}\end{cases}}$$

$$t=z+1,z+2,\ldots,z+b-b_h(z)-1$$
$$t=z+b-b_h(z),z+b-b_h(z)+1,\ldots,w$$

and consequently the probability to win the Bingo for the h-nth score-card is

<17> $$\boxed{\sum_{t=z+b-b_h(z)}^{w}p^{b}_{b_h|b_1,b_2,\ldots b_N}(z,t)=\sum_{t=z+b-b_h(z)}^{w}p^{b}_{b_h}(z,t)\prod_{j=1,j\neq h}^{N}(1-p^{b}_{b_j}(z,t))+}$$

$$\boxed{+\sum_{k=2}^{N}\sum_{h_2,h_3,\ldots,h_k}^{\binom{N-1}{k-1}}\frac{p^{b}_{b_h\wedge b_{h_2}\wedge\ldots\wedge b_{h_{k1}}|b_1,b_2,\ldots b_N}(z,t)}{k}=p^{b}_{b_h|b_1,b_2,\ldots b_N}(z,w)}\qquad h=1,2,\ldots,N$$

obviously resulting

$$\sum_{h=1}^{N}p^{b}_{b_h|b_1,b_2,\ldots b_N}(z,w)=1$$

Indicating with R=ROW the number of rows in a score-card and similarly to the process described before, the probability to obtain the Line win for the k-nth row in the h-nth score-card (playing with $l_{h,k}(z)<l$ checked numbers) is

<18> $$\boxed{p^{l}_{l_{h,k}|l_{1,1},l_{1,2},\ldots l_{N,R}}(z,w)}\qquad h=1,2,\ldots,N;k=1,2,\ldots R$$

having indicated with $l=LIN$ and being

$$\sum_{h=1}^{N}\sum_{k=1}^{R}p^{l}_{l_{h,k}|l_{1,1},l_{1,2},\ldots l_{N,R}}(z,w)=1$$

Starting from the initial price of each score-card participating to the game

<19> $$\boxed{PRI=PRC\cdot(ALPHA+BETA)}$$

the **current price** of the **h**-nth score-card (playing, after the drawing of z numbers, with $l_{h,k}(z)<l\mid k=1,2,\ldots,R$ checked numbers on the rows and $b_h(z)=\sum_{k=1}^{R}l_{h,k}<b$ checked numbers on the score-card) results

<20> $$\boxed{PRV_h=PRC\cdot NCA\cdot(ALPHA\cdot\pi^{b}_{b_h|b_1,b_2,\ldots b_N}(z,w)+BETA\cdot\sum_{k=1}^{R}\pi^{l}_{l_{h,k}|l_{1,1},l_{1,2},\ldots l_{N,R}}(z,w))}\qquad h=1,2,\ldots,N$$

4. Future and option pricing

The model for the calculation of the **current price,** shown in the last relation <20>, can be used for the calculation of the **future price** of a score-card with a certain situation of checked numbers (on the different rows) after a certain number of drawings. At the same time it can be calculated the **option price** that provides the right to buy or sell (call or put) the score-card at a pre-determined price. This can be obtained in the following way:

- starting from the overall situation of checked boxes for the whole set of score-cards after z drawings, it is necessary to build up the model describing the future evolution of the scenario,

- with reference to the future time for the exercise of the **future** or the **option** it is needed to estimate the random variable “future” current price for the observed score-card,

- on the base of the defined random variable, it is possible to calculate (for example, using the appropriate mean) the **future price** and the **option price** of the score-card,

- at the future time, with the evolved game status, the contracts can be executed or the following differences can be settled.

APPENDIX

The appendix describes the game simulation using the **Apl2Win/IBM**Ô programming environment.

The game structure is ruled from the function **RULE**, that enables to build up the game rules array through the following input parameters:

```
WUM        number of numbers to be drawn
ROW        number of rows in a score-card
LIN        number of checked box for the Line win
COL        vector of the column numbers range-witdth in a score
           -card, being æCOL the number of columns
```

and the set up of further game parameters:

```
NUM½+/COL       numbers of numbers in the Bingo basket
BIN½ROWõLIN     number of checked box for the Bingo win
SET½NUMöBIN     number of score-cards in a subset
```

```
[0]  RUL½WRL RULE COL;NUM;WUM;ROW;LIN;BIN;SET
[1]  (WUM ROW LIN)½WRL
[2]  SET½(NUM½+/COL)öBIN½ROWõLIN
[3]  RUL½NUM WUM ROW(,COL)LIN BIN SET
```

The set up of the playing score-cards can be done in two different ways:

- **a)** creation of a set of score-cards,
- **b)** creation of a set with a number of score-cards subsets.

The creation of the set **a)** is possible through the derivative function **(RUL CARDS)** obtained by applying the operator **CARDS** to the game rules array.

The function (**RUL CARDS**) uses the function **WCARD** (and the function **WWCARD)**.

```
[0]  CAR½(RUL CARDS)NUN
[1]  CAR½(ìæCAR),þCAR½WCARDþNUNæâRUL
```

```
[0]  WCA½WCARD RUL;NUM;ROW;COL;LIN;DRW
[1]  (NUM ROW COL LIN)½1 3 4 5ãþâRUL
[2]  WCA½DRW,þ1++/(DRW½NUM?NUM)ø.>+\COL
[3]  WCA½1ãþþ1ÇãWWCARD/(ROWæâì0),â,âWCA
[4]  WCA½(âþûþWCA)ÓþWCA
```

```
[0]   WCA½NUL WWCARD CAR;DRW;INT;MSK;DRZ
[1]   MSK½(INTìINT)=ìæINT½2ãþDRW½1ãCAR
[2]   MSK½MSK\LINòûü(MSK/INT)ãþâCOL
[3]   WCA½CAR,âDRZ½MSK/DRW
[4]   COL½COL-(ìæCOL)î2ãþDRZ
[5]   (1ãWCA)½(~MSK)/DRW
```

The creation of the set **b)** (corresponding to a number of score-cards equal to the number of subset multiplied for the number of score-cards contained in every subset) is possible through the derivative function **(RUL SETS)** obtained by the application of the **SETS** operator on the game rules array.

The function (**RUL SETS**) uses the function **WSETS** (and function **WCARD** (and function **WWCARD**)).

```
[0]   SER½(RUL SETS)NUN
[1]   SER½(ìæSER),þSER½ã,/WSETþNUNæâRUL
```

```
[0]   WSE½WSET RUL;ROW;SET
[1]   (3ãRUL)½õ/(ROW SET)½3 7ãþâRUL
[2]   WSE½(ROW/ìSET)âWCARD RUL
```

It is possible to determine the subset of score-cards participating to the game through the function **SUB**, that draws a subset of score-cards from the set built up through the function **(RUL CARDS)** or through the function **(RUL SETS)**.

The **SUB** function uses the **WONLY** and **WSORT** functions.

```
[0]   SUB½NCA SUB CAR
[1]   SUB½(WSORT WONLY NCA)ãþâCAR
```

```
[0]   WON½WONLY ARR
[1]   WON½((ARRìARR)=ìæARR)/ARR½,ARR
```

```
[0]   WSO½WSORT ARR
[1]   WSO½(âûARR)ÓARR½,ARR
```

It is possible to determine the subset of the score-cards participating to the game (randomly drawn) through the function **XSUB**, that performs the random drawing of a pre-defined number of score-cards from the set built up through the function **(RUL CARDS)** or the function **(RUL SETS)**.

The function **XSUB** uses the function **WSORT**.

```
[0]   SUB½NCA XSUB CAR
[1]   SUB½(WSORT NCA?æCAR)ãþâCAR
```

The **START** function prepares the execution of the game using the game rules array and the set of participating score-cards. The function defines the arrays that will contain the information on the game development (distribution of score-cards and lines in relation to the checked drawn numbers).

The **START** function uses the operator **XPOINT** (and operator **XDRAW** and function **WPOINT** (and function **WFREQ**)) and the function **DISP**.

```
[0]   RUL START CAR
[1]   DISP XPNT½((XRUL½RUL)XPOINT XCAR½CAR)0
```

```
[0]   PNT½(RUL XPOINT CAR)NDR;DRN;DRW;NUW
[1]   (DRN DRW)½(RUL XDRAW)NDR
[2]   PNT½(âþ(NUW½0,ìæDRW)ÆþâDRW)WPOINTþâCAR
[3]   PNT½(âDRN DRW),NUW,þPNT
```

```
[0]   DRW½(RUL XDRAW)NDR;NUM;WUM
[1]   (NUM WUM)½1 2ãþâRUL
[2]   DRW½NDR,â(NDR½NDR¾WUM)ÆWUM?NUM
```

```
[0]   WPN½DRW WPOINT CAR;LIN;BIN;PBN;PLN
[1]   (LIN BIN)½5 6ãþâRUL
[2]   PBN½+/þPLN½+/þþ(1ÇþCAR)îþþâDRW
[3]   WPN½(BIN WFREQ PBN),LIN WFREQ PLN
```

```
[0]   WFR½BLX WFREQ PNX
[1]   WFR½â+/(0,ìBLX)ø.=îPNX
```

```
[0]   DIS½DISP ARR
[1]   DIS½DISPLAYí(1,æARR)æARR
```

The game evolution is controllable through the **GAME** function that updates the arrays created with the **START** function on the base of the game rules after the drawing of a number.

The **GAME** function uses the operator **POINT** (and operator **WDRAW** (and function **WONLY**) and function **WPOINT** (and function **WFREQ**)) and the function **DISP**.

```
[0]   GAME NDR
[1]   DISP XPNT½XPNT(XRUL POINT XCAR)NDR
```

```
[0]   PNT½PNZ(RUL POINT CAR)NDR;DRN;DRW
[1]   (DRN DRW)½(1ãPNZ)(RUL WDRAW)NDR
[2]   PNT½(âDRN DRW),âDRN,(âDRW)WPOINT CAR
```

```
[0]   DRW½DRZ(RUL WDRAW)NDR;NUM
[1]   DRW½WONLY(2ã2æDRZ),NDR¾NUM½1ãRUL
[2]   DRW½(æDRW),âDRW½((NDRô0)õæDRW)æDRW
```

The game evolution can be already obtained automatically through the function **GAMEE**, that, like **GAME**, generates the arrays update after the drawing of a random number always in accordance to the current game rules.

The **GAMEE** function uses the **GAME** function (and operator **POINT** (and operator **WDRAW** (and function **WONLY**) and function **WPOINT** (and function **WFREQ**)) and function **DISP**).

```
[0]   GAMEE;NDR
[1]   GAME(?æNDR)ãNDR½(ì1ãXRUL)~2ã1ãXPNT
```

Finally the game evolution can be generated, automatically and globally, through the **XGAME** function that, starting from the game rules array and like **GAME** and **GAMEE**, performs the arrays update after a global random drawing of a fixed number of numbers.

The function **XGAME** uses the operator **XPOINT** (and the operator **XDRAW** and the function **WPOINT** (and function **WFREQ**)) and the function **DISP**.

```
[0]   XGAME NDR
[1]   DISP XPNT½(XRUL XPOINT XCAR)NDR
```

Assuming that the game is characterised from the following parameter definitions:

• number of the playing score-cards	***NCA***
• cost of a single score-card	***PRC***
• % of the gross jackpot for the Bingo winner	***ALPHA***
• % of the gross jackpot for the Line winner	***BETA***

the following values will result at the starting time:

• amount retained from the game organiser	***MNT½NCAõPRC***
• prize-money for the Bingo winner	***MNTõALPHA***
• prize-money for the Line winner	***MNTõBETA***
• initial price of each playing score-card	***PRCõ(ALPHA+BETA)***

During the game it is possible to calculate the current price of each score-card taking into account the number of checked boxes in every row of it and in every row of each score-card.

The problem could be faced in the analytical way, but, according to the function previously described, it is preferable to use the following numeric-simulative approach.

The existing game situation can be considered as follows:

- vector of the drawn numbers
- partition of the playing score-cards subset into equivalence classes, as result of the equality between the checked boxes vector (with ***ROW*** components) related to every score-card

- definition of the equivalence classes number and numerousness of each class

After the simulation of the following game development for a number of times “big enough” (until the end of the game and in any case not further the number ***WUM*** of foreseen drawings), it is possible to consider the following status:

- definition of the eventual Line or Bingo prize-money for every simulation and every equivalence class
- definition of the average prize-money for every equivalence class
- definition of the average prize-money for each score-card in every equivalence class, dividing the average prize-money of the equivalence class to the numerousness of the class

The current price of each score-card type can be considered equal to the average prize-money of the equivalence class in which the score card is included.

Bibliography

- A. Annibali/F. Bellini - Bingo Pricing: a Game Simulation and Evaluation using APL – APL 2002 Conference, Madrid

- J.A.Brown/S.Pakin/R.P.Polivka – APL2 at a glance – Prentice Hall

- N.D.Thompson/R.P.Polivka – Apl2 in depth – Springer Verlag

- W.Feller – An introduction to probability theory and its applic14tions – Wiley

- E. Parzen – Modern probability and its applications – Wiley

- A.Papoulis – Probability, random variables, stochastic processes – McGraw Hill

- R.Graham/D.E.Knuth/O.Patashnik – Concrete mathematics – Addison Wesley

- D.E.Knuth – The art of computer programming – Addison Wesley

- I.B.M. – Apl2 Programming: Language Reference

- I.B.M. – Apl2 Programming: User's Guide