Chapter 11
Computing Optimal Decision Strategies using the Infinity Computer: the case of Non-Archimedean Zero-Sum Games

Marco Cococcioni, Lorenzo Fiaschi, Luca Lambertini

Abstract As is well known, zero-sum games are appropriate instruments for the analysis of several issues across areas including economics, international relations and engineering, among others. In particular, the Nash equilibria of any two-player finite zero-sum game in mixed-strategies can be found solving a proper linear programming problem. This chapter investigates and solves non-Archimedean zero-sum games, i.e., games satisfying the zero-sum property allowing the payoffs to be infinite, finite and infinitesimal. Since any zero-sum game is coupled with a linear programming problem, the search for Nash equilibria of non-Archimedean games requires the optimization of a non-Archimedean linear programming problem whose peculiarity is to have the constraints matrix populated by both infinite and infinitesimal numbers. This fact leads to the implementation of a novel non-Archimedean version of the Simplex algorithm called Gross-Matrix-Simplex. Four numerical experiments served as test cases to verify the effectiveness and correctness of the new algorithm. Moreover, these studies helped in stressing the difference between numerical and symbolic calculations: indeed, the solution output by the Gross-Matrix Simplex is just an approximation of the true Nash equilibrium, but it still satisfies some properties which resemble the idea of a

Marco Cococcioni
Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino 1, Italy,
e-mail: marco.cococcioni@unipi.it

Lorenzo Fiaschi
Department of Information Engineering, University of Pisa, Largo Lucio Lazzarino 1, Pisa, Italy,
e-mail: lorenzo.fiaschi@phd.unipi.it

Luca Lambertini
Department of Economics, University of Bologna, Strada Maggiore 45, Bologna, Italy,
e-mail: luca.lambertini@unibo.it

non-Archimedean ε-Nash equilibrium. On the contrary, symbolic tools seem to be able to compute the "exact" solution, a fact which happens only on very simple benchmarks and at the price of its intelligibility. In the general case, nevertheless, they stuck as soon as the problem becomes a little more challenging, ending up to be of little help in practice, such as in real time computations. Some possible applications related to such non-Archimedean zero-sum games are also discussed.

11.1 Introduction

Non-Archimedean zero-sum games are a contact point between three distinct branches of Mathematics, namely Non-Archimedean Analysis (NAA), Linear Programming (LP) and Game Theory (GT). The former deals with algebraic structures lacking of the Archimedes' property [4, 39], which means they contain elements that are not finitely comparable, i.e., their elements are infinitely large or infinitely small with respect to each other. Two examples are Sergeyev's Grossone Methodology (GM) [42] (along with his patented Infinity Computer [21, 22, 40]) and Robinson's hyperreal numbers [38]: both allow one to use numbers which can be infinite, infinitesimal, rather than just finite as one may be used to work with.

LP is the well know and deeply studied branch of optimization theory firstly introduced by Kantorovič [28] and Dantzig [13]. It consists in the optimization of linear functions, subject to linear constraints (both equalities and inequalities) [47]. Moreover, LP has a very well know and strict relation with GT, i.e., the discipline which models the behaviour of rational agents within competitive environments, commonly called games [31]. Indicating with $\mathcal{ZG}$ the set of two-person finite zero-sum games in mixed-strategies [33], then any game in $\mathcal{ZG}$ can be always transformed in an LP problem [1] whose optima are the Nash equilibria of the original game. In particular, such optimization task has the game's payoffs matrix as the constraint matrix of the problem. The word "finite" means that the games model competitive environments where each player is endowed with a finite number of possible strategies, the label "zero-sum" refers to the fact that gains in the game are always exactly balanced by other players' losses, while "mixed-strategies" means that the players adopt each strategy they have according to a probability distribution.

So far, the GM has been used both in LP and in GT, but separately. The application of the GM to LP [7, 8, 16] gave birth to the very first generalization of the Simplex algorithm to non-Archimedean quantities

[6], namely the Gross-Simplex (G-Simplex in short). It has been the starting point for the developing of the Gross-Matrix Simplex used in this chapter [10]. Another fruitful result from the interaction between GM and LP is an enhancement of the Big-M method [3, 13, 46], namely the Infinitely-Big-M method (I-Big-M) [9].

On the other hand, GT mainly leveraged GM to model both non-Archimedean payoffs (infinite, finite or infinitesimal) and non-Archimedean probabilities (finite or infinitesimal) in the Prisoner's Dilemma [23, 24, 25]. Other applications of GM to GT involve games on graphs [11, 12] and infinite decision-making processes [37]. More broadly, GM has also been successfully applied to more general optimization problems ([18, 19, 26, 44]), to machine learning [2, 5] and to many other fields.

All in all, this chapter introduces a new algorithm, namely Gross-Matrix-Simplex (in brief GM-S), able to find Nash equilibria of games in $\mathcal{NZG}$, i.e., the non-Archimedean extension of $\mathcal{ZG}$. In particular, it is able to cope with non-Archimedean LP problems also having the constraint matrix and the unknowns vector filled with non-Archimedean numbers. The efficacy of the algorithm is tested on several experiments. Since GM-S outputs an arbitrarily precise approximation of a true Nash equilibrium of a game, a relevant part of this chapter is dedicated to study such approximation, to characterize it and to show its anti-exploitation properties, which resemble those of standard ε-Nash equilibria. Before continuing, it is important to remark that GM is independent from non-standard analysis, as shown in [43].

11.2 Zero-Sum Games

Any game $\mathcal{G}$ is typically represented by the triple $\{N, S, u\}$, where $N = \{1, \dots, n\}$ indicates the set of players which take part to the game. Each of them has at his/her disposal a set of strategies to adopt, namely S_i. The set S is defined as the Cartesian product of all the players sets of strategies, i.e., $S = \prod_{i=1}^{n} S_i$. Finally, $u : S \to \mathbb{R}^n$ is the game utility function, where the entry $u_i : S \to \mathbb{R}$ is the i-th player utility function, which assumes his/her income provided the strategies implemented by all the participants.

The game-theoretic model which will be generalized to the case of non-Archimedean quantities is the one of finite two-person zero-sum game in mixed-strategies. Formally, a zero-sum game [48] is any game satisfying the following property:

$$\sum_{i=1}^{n} u_i(s) = 0 \ \ \forall\, s \in S,$$

which means that the total utility experienced by the players altogether is always zero (whence the zero-sum label). On the other hand, a game is *finite* if the competitive scenario involves agents each having a finite number of strategies, i.e.,

$$|S_i| \in \mathbb{N} \ \ \forall i \in N.$$

Whenever the number of players is finite too, these games take the name of *matrix games*, since they are fully represented by the n-dimensional matrix $A = [u(s_k)]_k$, where $k = (k_1, \ldots, k_n)$, $s_k = (s_{k_1}, \ldots, s_{k_n})$, $k_i \in \{1, \ldots, |S_i|\}$ and $s_{k_i} \in S_i$, $i = 1, \ldots, n$. An interesting corner case consists of those games involving just two players, i.e., $n = 2$. Their matrix A simplifies a lot, ending to have the following structure:

$$A = \begin{bmatrix} u(s_1^1, s_1^2) & \cdots & u(s_1^1, s_{|S_2|}^2) \\ \vdots & \ddots & \vdots \\ u(s_{|S_1|}^1, s_1^2) & \cdots & u(s_{|S_1|}^1, s_{|S_2|}^2) \end{bmatrix},$$

where s_j^i means the j-th strategy of player i, i.e., $s_j \in S_i$, $i = 1, 2$. The label *mixed-strategies* refers to a game $G' = \{N, \Delta, h\}$ for which there exists a game $G = \{N, S, u\}$ and for which the following equalities are satisfied:

$$\Delta_{S_i} \coloneqq \left\{ \delta_i : S_i \to [0,1] \ : \ \sum_{s \in S_i} \delta_i(s) = 1 \right\}, \quad \Delta \coloneqq \prod_{i=1}^{n} \Delta_{S_i}$$

$$\delta : S \to [0,1], \quad \delta(s) = \prod_{i=1}^{n} \delta_i(s_i), \quad \delta_i \in \Delta_{S_i}, \quad s_i \in S_i$$

$$h_i(\delta) \coloneqq \sum_{s \in S} u_i(s)\delta(s), \quad h \coloneqq (h_1, \ldots, h_n).$$

As a result, $\mathcal{G}'$ describes the very same environment of $\mathcal{G}$, but this time each player can adopt a behavior δ_i which is a probability distribution over the strategies, rather than choosing a single one (whence the *mixed* nature of their strategies).

The choice of two-player finite zero-sum games in mixed-strategies is threefold:

- Finiteness and mixed-strategies guarantee the existence of at least one Nash equilibrium, as a corollary of the Nikaido-Isoda theorem [32];

- The presence of two players and the zero-sum property jointly imply that every Nash equilibrium is at the intersection of minimax (or maximin) strategies;
- Together, these four properties and their consequences allow one to reformulate the search for Nash equilibria as an LP problem [1].

Let one give a better look to the last dot. The Nash equilibria of games in $\mathcal{ZG}$ are all and only the mixed-strategies x and y which satisfy

$$\min_{x \in \Delta_{S_1}} \max_{y \in \Delta_{S_2}} x^T A y = \max_{y \in \Delta_{S_2}} \min_{x \in \Delta_{S_1}} x^T A y \,, \tag{11.1}$$

where A is the game matrix. The primal LP reformulation of (11.1) is the one in (11.2), while the dual is (11.3).

$$\begin{aligned} \max\ & \lambda \\ \text{s.t. } & \lambda \mathbf{1}^T \leq x^T A \\ & x^T \mathbf{1} = 1 \\ & x \geq 0 \end{aligned} \tag{11.2}$$

$$\begin{aligned} \min\ & \mu \\ \text{s.t. } & Ay \leq \mu \mathbf{1} \\ & \mathbf{1}^T y = 1 \\ & y \geq 0 \end{aligned} \tag{11.3}$$

where $\mathbf{1}$ is the vector, of proper dimension, filled by ones, and $\lambda, \mu \in \mathbb{R}$. Both the cases allow a rewriting into the canonical form reported below:

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. } & Ax = b \\ & x^T \mathbf{1} = 1 \\ & x \geq 0 \end{aligned}$$

Nash and ε-Nash equilibria in numerical algorithms

In most of the cases, numerical routines implemented to find Nash equilibria, e.g. Simplex algorithm for games in $\mathcal{ZG}$, output just an approximation of them. This is due to the rounding errors induced by the finiteness of the machine on the computations, which may be themselves ill-conditioned. Such errors propagate during the algorithm execution and persist in its output, playing a crucial role from a game theoretical perspective since the result may not form a Nash equilibrium anymore. This implies that unilateral deviation from that strategy profile may lead to benefits for the deviator. In finite games however, such a benefit is bounded, testifying that the approximated strategies form a ε-Nash equilibrium [35], a relaxed version of the Nash equilibrium.

Originally introduced in [36], ε-Nash equilibria are one of the key ingredients of Proposition 11.1, which links the deviator benefit bound to the quality of the optimal strategy approximation.

Definition 11.1 (Nash equilibrium)

Let $\mathcal{G}$ be a two-person zero-sum game and $H(x, y)$ be the game value function. Then, (x^*, y^*) is said to be a Nash equilibrium if and only if $H(x^*, y) \leq H(x^*, y^*) \leq H(x, y^*)\ \forall x \in \Delta_{S_1}, y \in \Delta_{S_2}$.

Definition 11.2 (ε-Nash equilibrium)

Let $\mathcal{G}$ be a two-person zero-sum game and $H(x, y)$ be the game value function. Then, (x^*, y^*) is said to be an ε-Nash equilibrium, indicated by $(x_\varepsilon, y_\varepsilon)$, if and only if $H(x^*, y) - \varepsilon \leq H(x^*, y^*) \leq H(x, y^*) + \varepsilon\ \forall x \in \Delta_{S_1}, y \in \Delta_{S_2}$, $\varepsilon \geq 0$.

Lemma 11.1 *Let $\mathcal{G}$ be a finite two-person zero-sum game and $H(x, y) = x^T Ay$ be the game value function. If (x^*, y^*) is a Nash equilibrium, then $\min\limits_i (Ay^*)_i = H(x^*, y^*) = \max\limits_j (x^{*T} A)_j$.*

Proof The proof comes straightforwardly from Definition 11.1; indeed, if there exists $\overline{j}$ such that $H(x^*, y^*) < (x^{*T} A)_{\overline{j}}$, then (x^*, y^*) is no more a Nash equilibrium. To prove it, let $\overline{y}$ be the deterministic strategy where $\overline{y}_j = 1$ if $j = \overline{j}$ and zero otherwise. Then, it happens that $H(x^*, y^*) = x^{*T} Ay^* < x^{*T} A\overline{y} = H(x^*, \overline{y}) \leq \max_j (x^{*T} A)_j$, which is in contrast with the assumption that $H(x^*, y^*)$ is a Nash equilibrium. □

Lemma 11.2 *Let $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times m}$. Then, $\max\limits_j (x^T A)_j \leq ||x||\,||A_x||$, where $||A_x|| = \sup\limits_{u\in\mathcal{S}^n} ||u^T A||$.*

Proof The following two inequalities are true and prove the lemma: $\max\limits_j (x^T A)_j \leq ||x^T A|| \leq ||x||\,||A_x||$. The first one holds by construction, while the second follows from the Cauchy-Schwarz inequality. □

Hereinafter, always assume $x \in \Delta_{S_1}$ and $y \in \Delta_{S_2}$. This shall increase the conciseness and improve the readability of the text.

Definition 11.3 (δ-approximating strategy)

Let x be an approximation of the strategy x^*. Then, x is said to δ-approximate x^*, indicated by x_δ, if and only if $||x^* - x_\delta|| \leq \frac{\delta}{2}$, $\delta \in \mathbb{R}^+$.

Proposition 11.1 *Let $\mathcal{G}$ be a finite two-person zero-sum game and $H(x, y) = x^T Ay$ be the game value function. Let also (x_δ, y_δ) be the δ-approximation of the game Nash equilibrium (x^*, y^*), i.e., $||x^* - x_\delta|| \leq \frac{\delta}{2}$ and $||y^* - y_\delta|| \leq \frac{\delta}{2}$. Then, (x_δ, y_δ) is an ε-Nash equilibrium with $\varepsilon = \delta||A||$.*

Proof Let $\Delta_x = x^* - x_\delta$ and $\Delta_y = y^* - y_\delta$. By construction and applying Lemma 11.2 to the last inequality, one has

$$\begin{aligned}
x_\delta^T A y_\delta \leq \max_y x_\delta^T A y = \max_j (x_\delta^T A)_j = \max_j ((x_\delta + \Delta_x - \Delta_x)^T A)_j = \\
= \max_j ((x^* - \Delta_x)^T A)_j = \max_j (x^{*T} A - \Delta_x^T A)_j \leq \max_j (x^{*T} A)_j - \min_j (\Delta_x^T A)_j \leq \\
\leq \max_j (x^{*T} A)_j + \frac{\delta}{2} ||A||,
\end{aligned} \tag{11.4}$$

where $||A|| = \max(||A||_x, ||A||_y)$. By means of analogous manipulations, it holds true that

$$\min_i (Ay^*)_i - \frac{\delta}{2} ||A|| \leq \min_i (Ay_\delta)_i \leq x_\delta^T A y_\delta.$$

Leveraging Lemma 11.1, one can rewrite it as

$$H(x^*, y^*) - \frac{1}{2}\varepsilon \leq \min_x H(x, y_\delta) \leq H(x_\delta, y_\delta) \leq \max_y H(x_\delta, y) \leq H(x^*, y^*) + \frac{1}{2}\varepsilon, \tag{11.5}$$

having imposed $\varepsilon = \delta ||A||$. The proof completes using the first and last inequalities of (11.5), which give

$$\begin{aligned}
H(x^*, y^*) - \frac{1}{2}\varepsilon \leq \min_x H(x, y_\delta) \Longrightarrow H(x^*, y^*) \leq \min_x H(x, y_\delta) + \frac{1}{2}\varepsilon \\
\max_y H(x_\delta, y) \leq H(x^*, y^*) + \frac{1}{2}\varepsilon \Longrightarrow \max_y H(x_\delta, y) - \frac{1}{2}\varepsilon \leq H(x^*, y^*)
\end{aligned}$$

and plugging them back into (11.5), respectively at the end and at the beginning of the inequalities chain, obtaining:

$$\max_y H(x_\delta, y) - \varepsilon \leq H(x_\delta, y_\delta) \leq \min_x H(x, y_\delta) + \varepsilon.$$

The interpretation of Proposition 11.1 is the following: in real-world standard applications, one can arbitrarily approximate the game value $H(x^*, y^*)$, i.e., up to a certain user-defined tolerance ε by opportunely setting δ in the optimizing algorithm (of course, both ε and δ must be greater than the machine precision). Indeed, δ measures the closeness of (x_δ, y_δ) to the truly optimal Nash-equilibrium and at the same time it bounds the value of ε. Therefore, when δ approaches zero also ε does, forcing the deviator's benefit towards values sufficiently small to be considered negligible for practical purposes.

11.3 Non-Archimedean Zero-Sum Games and the Gross-Matrix-Simplex Algorithm

A non-Archimedean zero-sum game is a zero-sum game where the payoffs can assume also infinite and infinitesimal values. A very intuitive method to represent them within a machine is leveraging Grossone Methodology [42], where each payoff $\tilde{z}$, which can be made by different infinite, finite and infinitesimal components, is represented as follows:

$$\tilde{z} = \sum_{i \in \mathbb{P}} z_i ①^i, \tag{11.6}$$

where $\mathbb{P} \subset \mathbb{Z}$ is a finite set of G-powers, $z_i \in \mathbb{R}\ \forall i \in \mathbb{P}$. As an example, consider the payoff $4① + 3 - 2①^{-1}$. One can interpret it as a reward constituted by three goods ordered by priority, approach which was originally proposed in [41]. In fact, it represents four units of the most important ware, three of the second one and a debt of two units of the third one. This is possible because the value of each good (a finite number) contributes to the payoff weighted by a scalar which is infinitely incommensurable with all the others. In the current case, the first and most important commodity has an infinite impact on the value of the payoff (indeed, it is multiplied by the infinite scalar ①), the second commodity has a finite effect (it is multiplied by $1 = ①^0$), while the third commodity an infinitesimal one.

Section 11.2 showed that one can find at least one Nash equilibrium for each game in $\mathcal{ZG}$ solving the LP problem coupled with it. In case games in $\mathcal{NZG}$ are considered, the LP problems turns into a non-Archimedean one (in brief, NALP). In literature, there already exists a Simplex-like algorithm able to deal with certain NALP problems, namely the Gross-Simplex algorithm [6]. It consists of an improved version of the Dantzig's algorithm enhanced by means of GM in order to cope with non-Archimedean cost functions. Nevertheless, its implementation is useless in the current context since the NALP problem associated to games in $\mathcal{NZG}$ have the constraints matrix A which is non-Archimedean.

This fact motivates the implementation of a further generalization of the Simplex algorithm, namely the Gross-Matrix-Simplex, which is able to solve such class of NALP and, as a consequence, to output strategy profiles which are themselves non-Archimedean. The problem to solve is the following:

$$\begin{aligned} \min \ & c^T \tilde{x} \\ \text{s.t.} \ & \tilde{A}\tilde{x} = b \\ & \tilde{x} \geq 0 \end{aligned} \tag{11.7}$$

Algorithm 1 The Gross-Matrix-Simplex algorithm

Step 0. The user has to provide the initial set $\mathtt{B}$ of basic indices.

Step 1. Compute $\tilde{x}_{\mathtt{B}}$ as: $\tilde{x}_{\mathtt{B}} = \tilde{A}_{\mathtt{B}}^{-1} b$ (where $\tilde{A}_{\mathtt{B}}$ is the sub-matrix obtained by $\tilde{A}$ by considering the columns indexed by $\mathtt{B}$, while $\tilde{A}_{\mathtt{B}}^{-1}$ is the inverse of $\tilde{A}_{\mathtt{B}}$, i.e., the non-Archimedean matrix which satisfies the equality $\tilde{A}_{\mathtt{B}}^{-1}\tilde{A}_{\mathtt{B}} = \tilde{A}_{\mathtt{B}}\tilde{A}_{\mathtt{B}}^{-1} = I$).

Step 2. Compute $\tilde{y}$ as: $\tilde{y} = c_{\mathtt{B}}^T \tilde{A}_{\mathtt{B}}^{-1}$ (where $\tilde{y}$ is a G-vector obtained by linearly combining the purely finite vector $c_{\mathtt{B}}^T$ by the G-scalar elements in the rows of $\tilde{A}_{\mathtt{B}}^{-1}$).

Step 3. Compute $\tilde{s}$ as: $\tilde{s} = c_{\mathtt{N}}^T - \tilde{y}\tilde{A}_{\mathtt{N}}$ (where $\mathtt{N}$ is the complementary set of $\mathtt{B}$) and then select the maximum (gradient rule). When this maximum is negative (being it infinite, finite or infinitesimal), then the current solution is optimal and the algorithm stops. Otherwise, the position of the maximum in the G-vector $\tilde{s}$ is the index k of the entering variable $\mathtt{N}(k)$.

Step 4. Compute $\tilde{d}$ as: $\tilde{d} = \tilde{A}_{\mathtt{B}}^{-1}\tilde{A}_{\mathtt{N}(k)}$, where $\tilde{A}_{\mathtt{N}(k)}$ is the G-vector corresponding to the $\mathtt{N}(k)$-th column of $\tilde{A}$.

Step 5. Find the largest G-scalar $\tilde{t} > 0$ such that $\tilde{x}_{\mathtt{B}} - \tilde{t}\tilde{d} \geqslant 0$. If there is not such a $\tilde{t}$, then the problem is unbounded (STOP); otherwise, at least one component of $\tilde{x}_{\mathtt{B}} - \tilde{t}\tilde{d}$, say h, equals to zero and the corresponding variable is the leaving variable.

Step 6. Update sets $\mathtt{B}$ and $\mathtt{N}$ by swapping $\mathtt{B}(h)$ and $\mathtt{N}(k)$, then return to Step 1.

where $\tilde{A}$ is the G-matrix containing the payoffs, c and b are the (purely finite) objective and constant-terms vectors.

Algorithm 1 reports the pseudocode of the Gross-Matrix-Simplex. A posteriori, one could have achieved a better coherency if the Gross-Simplex algorithm would have been named Gross-Cost-Simplex, since it is only able to handle non-Archimedean cost functions $\tilde{c}$. The most interesting instruction Algorithm 1 is the computation of the inverse of a non-Archimedean matrix, which is achieved by means of the Gaussian-Jordan elimination algorithm tailored to cope with non-Archimedean quantities. A very appealing way to improve the algorithm is to substitute the explicit computation of the inverse with an incremental LU factorization. This should lead to better numerical stability, as it happens in the original Simplex algorithm. A dedicated work about it seems perfectly reasonable and absolutely promising

The next subsection investigates a property similar to Proposition 11.1 for games in $\mathcal{NZG}$, i.e., the reliability of numerical solvers for Nash equilibria in case of non-Archimedean payoffs. Then, the discussion will move to show the effectiveness of GM-S to solve NALP problems, i.e., to numerically find Nash equilibria of games in $\mathcal{NZG}$. To do this, several test cases are presented, solved and discussed.

ε-Nash optimality in non-Archimedean contexts

Similarly to standard routines, algorithms working with non-Archimedean numbers suffer from rounding errors the G-digits: for instance, the machine represents the G-number $\frac{1}{3}①$ as $0.3333①$ (where the number of

3s is architecture-dependent). However, they are afflicted by one further source of inaccuracy: the finite space for G-scalars components, e.g., a computer cannot exactly represent the G-scalar $\frac{1}{①+1} = \frac{1}{①} - \frac{1}{①^2} + \frac{1}{①^3} + \ldots = \sum_{i \in \mathbb{N}} (-1)^{i-1} ①^{-i}$ because it would need an infinite memory. Therefore, it is interesting to investigate whether the approximated non-Archimedean Nash equilibria found by numerical routines are somehow related to a non-Archimedean version of ε-Nash equilibria, as it happens in the standard case (see Section 11.2). This time, the number of G-scalar components the machine is able to store, say k, shall play a role as well as the rounding G-digits tolerance δ. Proposition 11.2 formally extends the result in Proposition 11.1 to the non-Archimedean scenario. As before, the definitions of (δ, k)-approximating strategy and (ε, k)-Nash equilibrium are needed, i.e., the non-Archimedean extension of Definition 11.3 and Definition 11.2, respectively.

Roughly speaking, the former states that a non-Archimedean strategy $\tilde{x}$ (δ, k)-approximates $\tilde{x}^*$ if the G-digits of the first k components of $||\tilde{x}^* - \tilde{x}||$ (which is a G-scalar) are bounded by a value proportional to δ. Notice that (δ, k)-approximating strategies are exactly the type of solutions GM-S is designed to seek for and to output. The notion (ε, k)-Nash equilibrium works in a very similar way, and Proposition 11.2 shall prove that they are the type of Nash equilibrium output by the GM-S algorithm. For the sake of simplicity, hereinafter assume that $||\tilde{A}||$ is a finite number. This fact does not affect generality since the following scaling is always possible whenever $\tilde{A}$ does not satisfy it: $\tilde{A} \leftarrow \tilde{A}(||\tilde{A}||)^{-1}$.

Definition 11.4 ((δ, k)-approximating strategy)

Let $\tilde{x}$ be an approximation of the non-Archimedean strategy $\tilde{x}^*$. Then, $\tilde{x}$ is said to be a (δ, k)-approximation of $\tilde{x}^*$, indicated by $\tilde{x}^k_\delta$, if and only if $||\tilde{x}^* - \tilde{x}|| = ||\tilde{\Delta}_x|| = \sum_{i \in \mathbb{Z}_0^-} \Delta_i ①^i$ is such that $|\Delta_i| < \frac{\delta}{2} \in \mathbb{R}^+ \ \forall i = 0, \ldots, 1-k, \ k \in \mathbb{N}$.

Definition 11.5 ((ε, k)-Nash equilibrium)

Let $\mathcal{G}$ be a finite non-Archimedean two-person zero-sum game and $H(\tilde{x}, \tilde{y})$ be the game non-Archimedean value function. Also suppose that $H(\tilde{x}, \tilde{y})$ assumes at most finite values (if not shrink it so). Then, the feasible strategy profile $(\tilde{x}^*, \tilde{y}^*)$ is said to be an (ε, k)-Nash equilibrium, indicated by $(\tilde{x}^*, \tilde{y}^*)^k_\varepsilon$, if and only if $\max_{\tilde{y}} H(\tilde{x}^*, \tilde{y}) - \tilde{\varepsilon} \leq H(\tilde{x}^*, \tilde{y}^*) \leq \min_{\tilde{x}} H(\tilde{x}, \tilde{y}^*) + \tilde{\varepsilon}$, where $\tilde{\varepsilon} = \sum_{i \in \mathbb{Z}_0^-} \varepsilon_i ①^i \geq 0$, $\varepsilon \in \mathbb{R}_0^+$ and $|\varepsilon_i| \leq \varepsilon \ \forall i = 0, \ldots, 1-k, \ k \in \mathbb{N}$.

Finally, Proposition 11.2 shows that the strategy profile output by GM-S is an (ε, k)-Nash equilibrium. This come from the fact that GM-S is designed to output (δ, k)-approximating strategies, while Proposition 11.2 links (δ, k)-approximations of Nash equilibria strategies to the property of being (ε, k)-Nash equilibria.

Proposition 11.2 *Let $\mathcal{G}$ be a finite non-Archimedean two-person zero-sum game and $\tilde{A}$ is the non-Archimedean payoff matrix. If all the entries of $\tilde{A}$ have the form of* (11.6)*, then any (δ, k)-approximation $(\tilde{x}^k_\delta, \tilde{y}^k_\delta)$ of a Nash equilibrium $(\tilde{x}^*, \tilde{y}^*)$ is an (ε, k)-Nash equilibrium.*

Proof The result of Equation (11.4) holds true until the last inequality even in a non-Archimedean context, i.e.,

$$\tilde{x}^T_\delta \tilde{A} \tilde{y}_\delta \leq \max_j (\tilde{x}^{*T} \tilde{A})_j - \min_j (\tilde{\Delta}^T_x \tilde{A})_j.$$

To continue with the inequality chain, let one introduce two new G-scalars $\tilde{v}^k_x$ and $\tilde{v}^{-k}_x$ such that $\tilde{v}^k_x$ is the first k components of $||\tilde{x}^* - \tilde{x}^k_\delta|| = ||\tilde{\Delta}_x|| = \sum_{i \in \mathbb{Z}^-_0} \Delta_i ①^i$, i.e., $\tilde{v}^k_x = \sum_{i=0}^{1-k} \Delta_i ①^i$, while $\tilde{v}^{-k}_x$ is the remaining components of $||\tilde{\Delta}_x||$, i.e., $\tilde{v}^{-k}_x = ||\tilde{\Delta}_x|| - \tilde{v}^k_x$. Then, Leveraging Lemma 11.2 (which holds even in a non-Archimedean context) it holds true that $-\min_j (\tilde{\Delta}^T_x \tilde{A})_j \leq ||\tilde{\Delta}_x||\, ||A|| = ||\tilde{v}^k_x||\, ||A|| + ||\tilde{v}^{-k}_x||\, ||A|| \leq \frac{\tilde{\delta}}{2} ||\tilde{A}|| + ||\tilde{v}^{-k}_x||\, ||A|| = \frac{1}{2} \tilde{\varepsilon}^k_x + \frac{1}{2} \tilde{\varepsilon}^{-k}_x = \frac{1}{2} \tilde{\varepsilon}_x$, where $\tilde{\delta} = \sum_{j=0}^{1-k} \delta ①^j$, $\tilde{\varepsilon}^k_x = \delta ||\tilde{A}||$, $\tilde{\varepsilon}^{-k}_x = 2 ||\tilde{v}^{-k}_x||\, ||A||$. Finally, define $\tilde{\varepsilon}_y$ accordingly and set $\tilde{\varepsilon} = \max(\tilde{\varepsilon}_x, \tilde{\varepsilon}_y)$, i.e., $\tilde{\varepsilon} = \tilde{\varepsilon}^k + \tilde{\varepsilon}^{-k}$, where $\tilde{\varepsilon}^k = \tilde{\varepsilon}^k_x = \tilde{\varepsilon}^k_y$ and $\tilde{\varepsilon}^{-k} = \max(\tilde{\varepsilon}^{-k}_x, \tilde{\varepsilon}^{-k}_y)$. Reasoning similarly to Proposition 11.1, one obtains:

$$\max_{\tilde{y}} H(\tilde{x}^k_\delta, \tilde{y}) - \tilde{\varepsilon} \leq H(\tilde{x}^k_\delta, \tilde{y}^k_\delta) \leq \min_{\tilde{x}} H(\tilde{x}, \tilde{y}^k_\delta) + \tilde{\varepsilon}.$$

By definition, one can rewrite $\tilde{\varepsilon}^k$ as follows:

$$\tilde{\varepsilon}^k = \sum_{h=0}^{1-k} \varepsilon_h ①^h = \tilde{\delta} ||\tilde{A}|| = \left(\sum_{i=0}^{1-k} \delta ①^i \right) \left(\sum_{j \in \mathbb{P}} a_j ①^j \right) = \delta \sum_{i=0}^{1-k} \sum_{j \in \mathbb{P}} a_j ①^{i+j},$$
$$\varepsilon_h = \delta \sum_{(i,j) \in \mathcal{H}_h} a_j ①^{i+j} \quad \forall\, h = 0, \ldots, 1-k,$$

where $\mathcal{H}_h = \{(i, j) \in \{0, \ldots, 1-k\} \times \mathbb{P} \mid i + j = h\}$. Of course, $\mathcal{H}_h = \emptyset \Longrightarrow \varepsilon_h = 0$ by construction. In addition, it holds true that

$$|\varepsilon_h| = \delta \left| \sum_{(i,j) \in \mathcal{H}_h} a_j ①^{i+j} \right| \leq \delta \sum_{(i,j) \in \mathcal{H}_h} |a_j| ①^{i+j} \leq \delta \max_h |\mathcal{H}_h| \max_{j \in \mathbb{P}} |a_j| \leq \delta k \max_{j \in \mathbb{P}} |a_j| = \varepsilon,$$

Table 11.1 Rock-Paper-Scissors canonical form

$\wp_1$ \ $\wp_2$	**R**	**P**	**S**
R	0	1	-1
P	-1	0	1
S	1	-1	0

which means that the absolute value of all the G-digits of $\tilde{\varepsilon}^k$ are bounded by a positive constant depending on δ, similarly to Definition 11.5. The proof completes noticing that, by construction, $\tilde{\varepsilon}^k$ and $\tilde{\varepsilon}^{-k}$ are completely separated from components perspective, i.e., the smallest G-power in $\tilde{\varepsilon}^k$ is greater than the biggest one in $\tilde{\varepsilon}^{-k}$ (which comes from the particular choice of $\tilde{v}_x^k$ and $\tilde{v}_x^{-k}$). This implies that the first k components of $\tilde{\varepsilon}$ form exactly $\tilde{\varepsilon}^k$, and therefore $(\tilde{x}_\delta^k, \tilde{y}_\delta^k)$ satisfies Definition 11.5, i.e., it is an (ε, k)-Nash equilibrium. □

In fact, Proposition 11.2 states a finer the approximation of the optimal strategy profile corresponds with a higher negligiblity of the first k components of the deviation benefit. Notice that nothing can be said about the term $\tilde{\varepsilon}^{-k}$ except that its magnitude is not greater than ①$^{-k}$, which is reasonable since the machine representation truncates G-scalars up to the (k-1)-th component, i.e., the computations are not aware of the components with magnitude smaller or equal to ①$^{-k}$.

11.4 Numerical Illustrations

This section aims at assess the effectiveness of GM-S algorithm to find Nash equilibria of games in $\mathcal{NZG}$. Furthermore, Section 11.4.1 spends some lines to verify that GM-S outputs are both (δ, k)-approximations and (ε, k)-Nash equilibria. The majority of the experiments refer to non-Archimedean variations of the very well known zero-sum game called rock-paper-scissors. The choice of this model comes from the fact it is simple enough to make its non-Archimedean modifications easy to understand and predictable in terms of Nash equilibria. Table 11.1 reports its canonical form, while its mathematical formulation is reported in (11.8), where A indicates the payoffs matrix whose content coincides with Table 11.1.

$$\min_{x\in\Delta_1} \max_{y\in\Delta_2} x^T A y \tag{11.8}$$

$$\Delta_1 = \Delta_2 = \left\{ (\rho_1, \rho_2, \rho_3) \in \mathbb{R}^3 \,\middle|\, \sum_{i=1}^{3} \rho_i = 1,\ \rho_i \geq 0\ \forall\, i \right\}$$

In this game there exists only one Nash equilibrium, which is shown in Equation (11.9):

$$x^* = y^* = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right] \tag{11.9}$$

It is right to say that all the experiments outcomes have been double-checked verifying that they are *basic* Nash equilibria [45]. The procedure to do it is reported in Algorithm 2.

Algorithm 2 Procedure to double-check the results provided by the GM-S algorithm

step 0. Let (x^*, y^*) be a Nash equilibrium for a given n-dimensional game obtained somehow (in our case by GM-S).

Step 1. Indicate with $\mathcal{A}_x$ and $\mathcal{A}_y$ the index set of active strategies in x^* and y^*, respectively. This means that $i \in \mathcal{A}_z \Leftrightarrow z_i > 0, i \in \{1, \ldots, n\}, z = x, y$

Step 2. Define the active matrix B as the payoff matrix reduced to the row indexes in $\mathcal{A}_x$ and the column indexes in $\mathcal{A}_y$, i.e., $B \coloneqq A_{\mathcal{A}_x}^{\mathcal{A}_y}$.

Step 3. Define C and D such that $C_i = B^i - B^{i+1}\ \forall\, i = 1, \ldots, |\mathcal{A}_y| - 1$ and $C_{|\mathcal{A}_y|} = \mathbf{1}$, while for D holds true that $D_i = B_i - B_{i+1}\ \forall\, i = 1, \ldots, |\mathcal{A}_x| - 1$ and $D_{|\mathcal{A}_x|} = \mathbf{1}$.

Step 4. Verify that $Cx^* = e$ and $Dy^* = e$ hold true, where we have indicated with e the last (under the natural ordering) vector of the canonical base with proper dimension, i.e., $e = (0, \ldots, 0, 1)^T$.

11.4.1 Experiment 1: Infinitesimally perturbed rock-paper-scissors

The first experiment involves the very same problem of Table 11.1 with one single payoff infinitesimally perturbed, namely the $\tilde{A}_{2,1}$ (see Equation (11.10)). The main idea is to check the GM-S algorithm sensibility to infinitesimal changes in the matrix game. To improve the readability, the infinitesimal components shall be colored in blue. Since the original problem admits a unique Nash equilibrium, it is reasonable to expect that the new game's one is infinitely close to the former. Table 11.2 reports GM-S iterations executed during the optimization, while Equation (11.11) shows the new Nash equilibrium, which confirms the intuition of infinity closeness. However, GM-S algorithm is able to tell *exactly how much* the two are similar (up to the machine precision, of course).

Table 11.2 GM-S iterations for Game 1 of Equation (11.10): infinitesimally perturbed rock-paper-scissors game

It.	Base	$\tilde{x}$	$c^T\tilde{x}$
1	{4, 5, 6}	$\left[0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0\right]$	$-①$
2	{2, 5, 6}	$\left[0, \frac{1}{4} - \frac{1}{8}①^{-1} + \frac{1}{16}①^{-2}, 0, 0, \frac{1}{4} + \frac{1}{8}①^{-1} - \frac{1}{16}①^{-2}, \frac{1}{2}, 0, 0, 0\right]$	$-\frac{3}{4}① - \frac{3}{8} + \frac{3}{16}①^{-1} - \frac{1}{16}①^{-2}$
3	{2, 3, 6}	$\left[0, \frac{1}{3} - \frac{1}{9}①^{-1} + \frac{1}{27}①^{-2}, \frac{1}{6} + \frac{1}{9}①^{-1} - \frac{1}{27}①^{-2}, 0, 0, \frac{1}{2}, 0, 0, 0\right]$	$-\frac{1}{2}① - \frac{1}{2}$
4	{2, 3, 1}	$\left[\frac{1}{3}, \frac{1}{3} - \frac{1}{9}①^{-1} + \frac{1}{27}①^{-2}, \frac{1}{3} + \frac{1}{9}①^{-1} - \frac{1}{27}①^{-2}, 0, 0, 0, 0, 0, 0\right]$	-1

$$\tilde{A} = \begin{bmatrix} 0 & 1 & -1 \\ -1-①^{-1} & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \qquad (11.10)$$

$$\tilde{x}^k_\delta = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} - \frac{1}{9}①^{-1} + \frac{1}{27}①^{-2} \\ \frac{1}{3} + \frac{1}{9}①^{-1} - \frac{1}{27}①^{-2} \end{bmatrix}, \qquad \tilde{y}^k_\delta = \begin{bmatrix} \frac{1}{3} - \frac{1}{9}①^{-1} + \frac{1}{27}①^{-2} \\ \frac{1}{3} \\ \frac{1}{3} + \frac{1}{9}①^{-1} - \frac{1}{27}①^{-2} \end{bmatrix} \qquad (11.11)$$

The reason why in Table 11.2 the cost function starts from an infinite value and the length of vector $\tilde{x}$ is larger than expected is due to the use of I-Big-M method [9] as wrapper for the GM-S solver. It provides an easy way to retrieve a feasible starting basis adding a set of artificial variables to the problem and infinitely penalizing them by the term ① in the cost function (as pioneered in [16, 17]). The fact that at the end of the optimization the cost function has a finite value means that all the artificial variables have exited the basis and the solution is a feasible one.

Finally, let one verify that the $(\delta, 3)$-approximating strategy profile $(\tilde{x}^k_\delta, \tilde{y}^k_\delta)$ is really an $(\varepsilon, 3)$-Nash equilibrium. Theoretically, the strategy pair $(\tilde{x}^k_\delta, \tilde{y}^k_\delta)$ approximates the Nash equilibrium $(\tilde{x}^*, \tilde{y}^*)$ defined as

$$\tilde{x}^* = \left[\frac{1}{3}, \frac{①}{3①+1}, \frac{3①+2}{3(3①+1)}\right]^T, \qquad \tilde{y}^* = \left[\frac{①}{3①+1}, \frac{1}{3}, \frac{3①+2}{3(3①+1)}\right]^T.$$

Expanding the second entry of $\tilde{x}^*$ one gets $\frac{①}{3①+1} = \sum_{i \in \mathbb{Z}_0^-} \frac{1}{3(-3)^{-i}} ①^i = \frac{1}{3} - \frac{1}{9}①^{-1} + \frac{1}{27}①^{-2} - \frac{1}{81}①^{-3} + \ldots$, which becomes exactly the second entry of $\tilde{x}^k_\delta$ when it is truncated up to the second infinitesimal component ($k = 3$). Repeating the same operation with all the other entries, the fact that $\tilde{x}^k_\delta$ and $\tilde{y}^k_\delta$ are $(\delta, 3)$-approximations of $\tilde{x}^*$ and $\tilde{y}^*$ is verified. Then, Proposition 11.2 states that if $(\tilde{x}^k_\delta, \tilde{y}^k_\delta)$ is a $(\varepsilon, 3)$-Nash equilibrium, then it must hold true that $|H(\tilde{x}^k_\delta, \tilde{y}^k_\delta) - \min_{\tilde{x}}(\tilde{x}, \tilde{y}^k_\delta)| \leq \tilde{\varepsilon} = \tilde{\varepsilon}^k + \tilde{\varepsilon}^{-k} = \tilde{\varepsilon}^{-k} < M①^{-k}$, with $M \in \mathbb{R}^+$ sufficiently big (the last equality comes from the fact that $\delta = 0$). The game value associated to (11.11) is $H(\tilde{x}^k_\delta, \tilde{y}^k_\delta) = \sum_{i=-1}^{-5} \frac{1}{3(-3)^{-i}} ①^i$, while the row player optimal strategy as answer to $\tilde{y}^k_\delta$ is$\bar{x} = [0, 1, 0]^T$, which leads to $H(\bar{x}, \tilde{y}^k_\delta) = -\frac{1}{9}①^{-1} + \frac{1}{27}①^{-2} - \frac{1}{27}①^{-3}$. Therefore, the deviation benefit is $|H(\tilde{x}^k_\delta, \tilde{y}^k_\delta) - H(\bar{x}, \tilde{y}^k_\delta)| = \frac{2}{81}①^{-3} + \frac{1}{243}①^{-4} - \frac{1}{729}①^{-5} = \varepsilon^{-k} < M①^{-3}$ for $M > \frac{2}{81}$. This means that $(\tilde{x}^k_\delta, \tilde{y}^k_\delta)$ is a $(\varepsilon, 3)$-Nash equilibrium since $\tilde{\varepsilon}^k$ is negligible and $\tilde{\varepsilon}^{-k}$ has magnitude not greater than $①^{-3}$ (provided that the same verification is done for column player as deviator, study which is omitted for brevity). In addition, notice how close the approximation of $H(\tilde{x}^k_\delta, \tilde{y}^k_\delta)$ to $H(\tilde{x}^*, \tilde{y}^*)$ is. Since $H(\tilde{x}^*, \tilde{y}^*) = -\frac{1}{3(3①+1)} = \sum_{i\in\mathbb{Z}^-} \frac{1}{3(-3)^{-i}}①^i = -\frac{1}{9}①^{-1} + \frac{1}{27}①^{-2} - \frac{1}{81}①^{-3} + \dots$, we have $|H(\tilde{x}^*, \tilde{y}^*) - H(\tilde{x}^k_\delta, \tilde{y}^k_\delta)| = |\sum_{\substack{i\in\mathbb{Z}\\ i<-5}} \frac{1}{3(-3)^{-i}}①^i|$: the game value is well approximated up to the fifth order of infinitesimal.

11.4.2 Experiment 2: A purely finite 4-by-3 game

In games where more than one Nash equilibrium exists, even an infinitesimal perturbation can significantly affect the optimal strategy choice. For instance, consider the game in Equation (11.12) which is built adding one strategy for the row player to the rock-paper-scissors of Table 11.8. The crucial aspect is that the new game has infinitely many Nash equilibria, without any further criterion by means of which to choose among them. Therefore, the choice of which mixed-strategy to play depends only on the actual implementation of the decision making algorithm, e.g., the Simplex algorithm.

$$A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \\ \frac{1}{2} & -1 & \frac{1}{2} \end{bmatrix} \tag{11.12}$$

Table 11.3 reports the iterations of GM-S algorithm running on the degenerate problem (11.12). The routine suggests that the row-player adopts the strategy $x^k_\delta = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0]^T$, which is essentially the same as in the original game (11.9), i.e., $x^* = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$. To improve the readability, the optimal strategies have been reported as fractions. Next subsection shall show that an infinitesimal perturbation of these payoffs causes a finite change on the equilibrium rather than an infinitesimal one.

Table 11.3 GM-S iterations for the purely finite Game 2 of Equation (11.12)

It.	Base	$\tilde{x}$	$c^T \tilde{x}$
1	{5, 6, 7}	$\left[0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0\right]$	$-①$
2	{5, 6, 1}	$\left[\frac{1}{4}, 0, 0, 0, \frac{1}{4}, \frac{1}{2}, 0, 0, 0, 0\right]$	$-\frac{3}{4}① - \frac{1}{4}$
3	{2, 6, 1}	$\left[\frac{1}{3}, \frac{1}{6}, 0, 0, 0, \frac{1}{2}, 0, 0, 0, 0\right]$	$-\frac{1}{2}① - \frac{1}{2}$
4	{2, 3, 1}	$\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, 0, 0, 0, 0\right]$	-1

11.4.3 Experiment 3: Infinitesimally perturbed 4-by-3 game

Assume now that a secondary information to better discriminate among the infinite Nash equilibria in (11.12) is available. It consists of a second payoff matrix A_s whose importance is subordinated to the one of Equation (11.12):

$$A_s = \begin{bmatrix} 0 & 2 & -2 \\ -2 & 0 & 2 \\ 2 & -2 & 0 \\ -1 & -2 & 1 \end{bmatrix}$$

As discussed at the beginning of Section 11.3 and shown in Equation (11.6), the two sources of information can be eventually merged without losing the priority information by means of GM. In particular, a new non-Archimedean zero-sum game is built summing the payoffs entry-wise and scaling down the ones having lesser priority by the $①^{-1}$, i.e., the new payoff matrix $\tilde{A}$ is obtained as

$$\tilde{A} = A + A_s ①^{-1},$$

which in fact means

$$\tilde{A} = \begin{bmatrix} 0 & 1 + 2①^{-1} & -1 - 2①^{-1} \\ -1 - 2①^{-1} & 0 & 1 + 2①^{-1} \\ 1 + 2①^{-1} & -1 - 2①^{-1} & 0 \\ \frac{1}{2} - ①^{-1} & -1 - 2①^{-1} & \frac{1}{2} + ①^{-1} \end{bmatrix}. \tag{11.13}$$

The overall game is now a non-Archimedean one, falling in the set $\mathcal{NZG}$, since $\tilde{A}$ is non-Archimedean.

In the face of the infinitesimal perturbation on the payoffs matrix induced by the secondary information, the result is all but negligible. As opposed to the problem in Table 11.1, there exist infinitely many Nash equilibria in (11.12). This means that the presence of a secondary payoffs

Table 11.4 GM-S iterations for Game 3 of Equation (11.13): row-player

It.	Base	$\tilde{x}$	$c^T \tilde{x}$
1	{5, 6, 7}	$0, 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, 0, 0$	$-\textcircled{1}$
2	{5, 4, 7}	$0, 0, 0, \frac{1}{4} - \frac{1}{4}\textcircled{1}^{-1} + \frac{1}{4}\textcircled{1}^{-2}, \frac{3}{8} - \frac{1}{8}\textcircled{1}^{-1} + \frac{1}{8}\textcircled{1}^{-2}, 0, \frac{3}{8} + \frac{3}{8}\textcircled{1}^{-1} - \frac{3}{8}\textcircled{1}^{-2}, 0, 0, 0$	$-\frac{3}{4}\textcircled{1} - \frac{1}{2} + \frac{1}{2}\textcircled{1}^{-1} - \frac{1}{4}\textcircled{1}^{-2}$
3	{5, 4, 1}	$\frac{3}{10} + \frac{3}{50}\textcircled{1}^{-1} + \frac{3}{250}\textcircled{1}^{-2}, 0, 0, \frac{2}{5} + \frac{2}{25}\textcircled{1}^{-1} + \frac{2}{125}\textcircled{1}^{-2}, \frac{3}{10} - \frac{7}{50}\textcircled{1}^{-1} - \frac{7}{250}\textcircled{1}^{-2}, 0, 0, 0, 0, 0$	$-\frac{3}{10}\textcircled{1} - \frac{14}{25} - \frac{14}{125}\textcircled{1}^{-1} - \frac{3}{250}\textcircled{1}^{-2}$
4	{2, 4, 1}	$\frac{2}{5} + \frac{4}{75}\textcircled{1}^{-1} - \frac{16}{25}\textcircled{1}^{-2}, \frac{1}{5} - \frac{28}{75}\textcircled{1}^{-1} + \frac{56}{125}\textcircled{1}^{-2}, 0, \frac{2}{5} + \frac{8}{25}\textcircled{1}^{-1} - \frac{48}{125}\textcircled{1}^{-2}, 0, 0, 0, 0, 0, 0$	-1

Table 11.5 GM-S iterations for Game 3 of Equation (11.13): column-player

It.	Base	$\tilde{y}$	$c^T \tilde{y}$
1	{4, 5, 6, 7}	$0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, 0, 0, 0, 0$	$-\textcircled{1}$
2	{4, 3, 6, 7}	$0, 0, \frac{2}{9} - \frac{17}{5}\textcircled{1}^{-1} + \frac{27}{50}\textcircled{1}^{-2}, \frac{4}{9} + \frac{1}{5}\textcircled{1}^{-1} - \frac{31}{100}\textcircled{1}^{-2}, 0, \frac{2}{9} + \frac{1}{10}\textcircled{1}^{-1} - \frac{3}{20}\textcircled{1}^{-2}, \frac{1}{9} + \frac{1}{25}\textcircled{1}^{-1} - \frac{2}{25}\textcircled{1}^{-2}, 0, 0, 0, 0$	$-\frac{7}{9}\textcircled{1} + \frac{253}{450} + \frac{394}{100}\textcircled{1}^{-1} - \frac{27}{50}\textcircled{1}^{-2}$
3	{4, 3, 6, 1}	$\frac{1}{10} + \frac{1}{25}\textcircled{1}^{-1} - \frac{8}{125}\textcircled{1}^{-2}, 0, \frac{3}{10} - \frac{7}{25}\textcircled{1}^{-1} + \frac{56}{125}\textcircled{1}^{-2}, \frac{1}{2} + \frac{2}{5}\textcircled{1}^{-1} - \frac{6}{25}\textcircled{1}^{-2}, 0, \frac{1}{10} - \frac{4}{25}\textcircled{1}^{-1} - \frac{18}{125}\textcircled{1}^{-2}, 0, 0, 0, 0, 0$	$-\frac{3}{5}\textcircled{1} - \frac{16}{25} + \frac{78}{125}\textcircled{1}^{-1} - \frac{48}{125}\textcircled{1}^{-2}$
4	{4, 3, 2, 1}	$\frac{1}{3} - \frac{4}{3}\textcircled{1}^{-1} + \frac{16}{3}\textcircled{1}^{-2}, \frac{1}{3} - \frac{8}{3}\textcircled{1}^{-1} + \frac{40}{3}\textcircled{1}^{-2}, \frac{1}{3} - \frac{8}{3}\textcircled{1}^{-2}, 4\textcircled{1}^{-1} - 16\textcircled{1}^{-2}, 0, 0, 0, 0, 0, 0, 0$	$-5 + 20\textcircled{1}^{-1} - 16\textcircled{1}^{-2}$
5	{7, 3, 2, 1}	$\frac{1}{3} - \frac{2}{9}\textcircled{1}^{-1} + \frac{4}{27}\textcircled{1}^{-2}, \frac{1}{3} - \frac{2}{9}\textcircled{1}^{-1} + \frac{4}{27}\textcircled{1}^{-2}, \frac{1}{3} - \frac{2}{9}\textcircled{1}^{-1} + \frac{4}{27}\textcircled{1}^{-2}, 0, 0, 0, \frac{2}{3}\textcircled{1}^{-1} - \frac{4}{9}\textcircled{1}^{-2}, 0, 0, 0, 0$	$-\frac{5}{3} + \frac{10}{9}\textcircled{1}^{-1} - \frac{4}{9}\textcircled{1}^{-2}$
6	{10, 3, 2, 1}	$\frac{1}{3} - \frac{8}{15}\textcircled{1}^{-2}, \frac{1}{3} - \frac{8}{15}\textcircled{1}^{-1} + \frac{8}{15}\textcircled{1}^{-2}, \frac{1}{3} - \frac{8}{39}\textcircled{1}^{-1}, 0, 0, 0, 0, 0, 0, \frac{4}{5}\textcircled{1}^{-1}, 0$	$-1 + \frac{4}{5}\textcircled{1}^{-1}$

in A_s informs the optimization algorithm of an additional discriminating rule to better single out the most preferable Nash equilibria. More importantly, the equilibrium found in Table 11.3 may not be optimal also with respect to the secondary information, while all those the strategy profiles which are may notably differ from it. This is the case of problem in (11.13), as testified by Table 11.4 and Table 11.5 which report the GM-S algorithm iterations computed to solve it for the row and the column-player, respectively.
The new equilibrium is:

$$\tilde{x}^*_\delta = \begin{bmatrix} \frac{2}{5} + \frac{4}{75}\textcircled{1}^{-1} - \frac{16}{25}\textcircled{1}^{-2} \\ \frac{1}{5} - \frac{28}{75}\textcircled{1}^{-1} + \frac{56}{125}\textcircled{1}^{-2} \\ 0 \\ \frac{2}{5} + \frac{8}{25}\textcircled{1}^{-1} - \frac{28}{125}\textcircled{1}^{-2} \end{bmatrix}, \qquad \tilde{y}^*_\delta = \begin{bmatrix} \frac{1}{3} + \frac{4}{15}\textcircled{1}^{-1} - \frac{8}{25}\textcircled{1}^{-2} \\ \frac{1}{3} - \frac{4}{15}\textcircled{1}^{-1} + \frac{8}{25}\textcircled{1}^{-2} \\ \frac{1}{3} \end{bmatrix}. \tag{11.14}$$

Observe how its finite part has changed from $x^k_\delta = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0]^T$ of the previous equilibrium to $x^k_\delta = [\frac{2}{5}, \frac{1}{5}, 0, \frac{2}{5}]^T$ of the current one.

11.4.4 Experiment 4: high dimensional games

This experiment aims at challenging GM-S algorithm on high-dimensional games, comparing the efficacy with symbolic approaches one. The first test case is not a true high-dimensional game since it involves only 8 strategies per player. However, it is still a significant experiment since it

$$
\begin{array}{l}
(-3.58904 + 103.344\,g + 720.946\,g^2 - 9651.52\,g^3 - 338238.\,g^4 - 3.18695\times10^6\,g^5 + 1.24018\times10^7\,g^6 + 2.8101\times10^8\,g^7 + \\
\quad 1.72011\times10^9\,g^8 + 6.24568\times10^9\,g^9 + 1.47068\times10^{10}\,g^{10} + 1.81561\times10^{10}\,g^{11} - 1.86775\times10^{10}\,g^{12} - 1.728\times10^{11}\,g^{13} - \\
\quad 5.42735\times10^{11}\,g^{14} - 1.15499\times10^{12}\,g^{15} - 1.77031\times10^{12}\,g^{16} - 1.66508\times10^{12}\,g^{17} + 4.72222\times10^{11}\,g^{18} + 6.3175\times10^{12}\,g^{19} + \\
\quad 1.73155\times10^{13}\,g^{20} + 3.38623\times10^{13}\,g^{21} + 5.46924\times10^{13}\,g^{22} + 7.68805\times10^{13}\,g^{23} + 9.65928\times10^{13}\,g^{24} + 1.10328\times10^{14}\,g^{25} + \\
\quad 1.16008\times10^{14}\,g^{26} + 1.13484\times10^{14}\,g^{27} + 1.04198\times10^{14}\,g^{28} + 9.04283\times10^{13}\,g^{29} + 7.44483\times10^{13}\,g^{30} + 5.81056\times10^{13}\,g^{31} + \\
\quad 4.27046\times10^{13}\,g^{32} + 2.91627\times10^{13}\,g^{33} + 1.80782\times10^{13}\,g^{34} + 9.75863\times10^{12}\,g^{35} + 4.15519\times10^{12}\,g^{36} + \\
\quad 8.94097\times10^{11}\,g^{37} - 6.2536\times10^{11}\,g^{38} - 1.05118\times10^{12}\,g^{39} - 9.33968\times10^{11}\,g^{40} - 6.43649\times10^{11}\,g^{41} - 3.73419\times10^{11}\,g^{42} - \\
\quad 1.87453\times10^{11}\,g^{43} - 8.23227\times10^{10}\,g^{44} - 3.16971\times10^{10}\,g^{45} - 1.07216\times10^{10}\,g^{46} - 3.19584\times10^9\,g^{47} - 8.59424\times10^8\,g^{48} - \\
\quad 2.18065\times10^8\,g^{49} - 5.66965\times10^7\,g^{50} - 1.49953\times10^7\,g^{51} - 3.77275\times10^6\,g^{52} - 694785.\,g^{53} - 78327.8\,g^{54} + \\
\quad 6045.51\,g^{55} - 923.935\,g^{56} - 1043.41\,g^{57} - 443.538\,g^{58} - 16.0318\,g^{59} + 30.3366\,g^{60} - 5.42663\,g^{61} + 0.175217\,g^{62}) / \\
((0.993515 + 0.482686\,g + 1.\,g^2)\,(3.27553 + 3.44271\,g + 4.46498\,g^2 + 1.31183\,g^3 + 1.\,g^4) \\
\quad (-0.780488 + 16.1408\,g + 72.0581\,g^2 + 80.2571\,g^3 + 101.998\,g^4 + 34.5331\,g^5 + 18.3454\,g^6 - 7.65728\,g^7 + 1.\,g^8) \\
\quad (-1.18922 - 13.4137\,g - 182.973\,g^2 - 267.794\,g^3 - 95.7229\,g^4 + 702.883\,g^5 + 1889.16\,g^6 + 2617.11\,g^7 + 2567.49\,g^8 + \\
\qquad 1513.39\,g^9 + 483.063\,g^{10} - 235.831\,g^{11} - 342.028\,g^{12} - 225.883\,g^{13} - 84.4367\,g^{14} - 20.2516\,g^{15} + 1.\,g^{16}) \\
\quad (-1.20679 + 22.8092\,g + 169.934\,g^2 - 117.669\,g^3 + 1673.5\,g^4 + 20154.3\,g^5 + 96848.8\,g^6 + 359978.\,g^7 + \\
\qquad 970133.\,g^8 + 2.21868\times10^6\,g^9 + 4.12681\times10^6\,g^{10} + 6.67195\times10^6\,g^{11} + 9.10681\times10^6\,g^{12} + 1.08871\times10^7\,g^{13} + \\
\qquad 1.1076\times10^7\,g^{14} + 9.80718\times10^6\,g^{15} + 7.23746\times10^6\,g^{16} + 4.47\times10^6\,g^{17} + 2.04326\times10^6\,g^{18} + \\
\qquad 558717.\,g^{19} - 163762.\,g^{20} - 297648.\,g^{21} - 227335.\,g^{22} - 106240.\,g^{23} - 36877.5\,g^{24} - 5395.33\,g^{25} + \\
\qquad 983.216\,g^{26} + 1276.05\,g^{27} + 343.541\,g^{28} + 53.5686\,g^{29} - 10.4184\,g^{30} - 2.82982\,g^{31} + 1.\,g^{32})),
\end{array}
$$

Fig. 11.1 First entry of x^* after symbolic computations in Mathematica (g here stands for ①). Please observe how the provided solution is very difficult to read.

already allows one to stress the limits of symbolic tools, such as Mathematica, which start to struggle even in this relatively low dimensional scenario. When game complexity grows, the gap between numerical and symbolic algorithms efficacy becomes more and more evident. The main reason is that, considering games in $\mathcal{NZG}$, symbolic tools search for Nash equilibria repeatedly applying Algorithm 2 with randomly chosen active strategies, as stated in [45], which is an unpractical and combinatorial NP-hard task. Moreover, when found, the Nash equilibrium readability would probably be quite low, as stressed by Figure 11.1 which reports the first entry of the row player optimal strategy computed in Mathematica (the letter g stands for ①). On the other hand, the numerical GM-S algorithm is able to show the approximated equilibrium components in a very interpretable way, e.g., Equation (11.15) where the entire solution of the 8×8 problem is shown. The payoff matrix and GM-S iterations can be found in the appendix of [10].

$$\tilde{x}^*_\delta = \begin{bmatrix} 0.18 - 0.36①^{-1} + 1.05①^{-2} \\ 0.15 + 0.01①^{-1} - 1.96①^{-2} \\ 0.26 - 0.66①^{-1} + 1.34①^{-2} \\ 0 \\ 0.22 + 0.3①^{-1} + 2.09①^{-2} \\ 0.08 + 0.36①^{-1} - 0.95①^{-2} \\ 0.11 + 0.34①^{-1} - 1.57①^{-2} \\ 0 \end{bmatrix}, \qquad \tilde{y}^*_\delta = \begin{bmatrix} 0.01 + 0.54①^{-1} + 3.12①^{-2} \\ 0.14 - 0.15①^{-1} - 1.03①^{-2} \\ 0 \\ 0.21 + 0.37①^{-1} + 2.71①^{-2} \\ 0.27 + 0.34①^{-1} - 0.81①^{-2} \\ 0 \\ 0.1 - 0.81①^{-1} - 2.19①^{-2} \\ 0.27 - 0.29①^{-1} - 1.8①^{-2} \end{bmatrix} \tag{11.15}$$

Considering a game with a 10×10 payoff matrix instead, the symbolic tool struggles a lot in computing the game solution and its execution time grows drastically, passing from 1.28 seconds of the 8×8 game to 6.98. To stress even more this fact, a true high-dimensional non-Archimedean zero-sum game has been considered. This time, each player can choose among 60 different strategies, and the game has 5 levels of priority information, i.e., 60×60 payoff matrix is filled by random G-scalars having 4 infinitesimal components. While Mathematica does not output the result in a reasonable amount of time for practical purposes, GM-S took only 122.487 seconds to compute the Nash Equilibrium, a quite remarkable result. For the sake of brevity, the matrix and the associated Nash equilibrium for both the 10×10 and 60×60 games are omitted.

11.5 A Brief Overview of Applications

As in [10], to which we refer the reader interested in a more detailed illustration, also here we offer a compact perspective of the models and fields which plausibly feature the non-Archimedean zero-sum property. Whenever the latter is identified, the game under examination can be treated using GM and of course the GM-S algorithm.

The first example has a dual nature, depending on whether players are firms or political parties, and is traditionally known as the Hotelling/Downs game of spatial competition [20, 27].

In both cases, players are supposed to choose their respective locations along a finite linear segment along which consumers or voters are distributed. In absence of a price mechanism, this can describe the choice of electoral platforms, with parties trying to acquire political consensus. In such a case the game has a zero-sum nature because any additional vote gained by a party is lost by the other (leaving aside any degree of abstention), and the model naturally lends itself to a non-Archimedean interpretation.

If instead the game features a price stage appearing after the location choice and prices are strategically set by firms, the game is no longer a zero-sum one. Yet, it becomes so if prices are regulated, in which case firms' profits depend solely on locations and the resulting relative size of market shares. It is also worth stressing that in such a case the Hotelling version of the model is observationally equivalent to Downs's.

This example is particularly relevant as it concerns a game which simultaneously belongs to the backbone of large literatures in industrial economics and politics. But several other non-Archimedean zero-sum games can be figured out.

For instance, another setting which has a twofold interpretation is that in which an entrepreneur owning a firm in dire straits is forced to sell it but maintains a symbolic amount of share and the family name on the gate of the production plant. From a realistic point of view, this payoff may look almost immaterial, but this does not correspond to the original owner's perception. The same applies to a seemingly different scenario in which we may just replace property with sovereignty. This holds for very small States surrounded by larger ones and embedded in fully integrated regions like the EU (think of Montecarlo, Liechtenstein, Andorra and San Marino). Political independence, accompanied by economic integration, may still matter a lot, although these Countries are net importers and, say, their basketball or football clubs may happen to play in the leagues of larger European Countries.

One additional example relies on [34] and is also connected to Patrolling games. Here, the matter is about construction operation purposes. The non-Archimedean property arises if the quality criteria adopted to decide which project to pursue are defined in binary terms, in such a way that the presence of any relevant component (say, fire escapes) is valued 1 while its absence is valued 0.

11.6 Conclusions

The chapter introduced for the first time the non-Archimedean zero-sum games and a numerical algorithm able to compute approximations of their Nash equilibria, namely the GM-S algorithm. A lot of attention has been dedicated to characterized the type of strategy profile the algorithm is able to output, showing a strong correlation with the standard

ε-Nash equilibrium. Both the algorithm implementation and the theoretical study leveraged Grossone Methodology to better reason about non-Archimedean quantities as well as execute numerical non-Archimedean operations. The chapter also illustrated several test cases, some of which high-dimensional, to verify the effectiveness of GM-S algorithm and to highlight the limits of symbolic tools. All the computations run over an Infinity Computer simulator implemented in software. Finally, Some possible real-world applications of this new modeling tool have been presented and discussed. It is right to say that such results would not be obtained without the previous achievements in non-Archimedean linear programming [6, 7, 9] and non-Archimedean Prisoner's Dilemmas [23, 24, 25].

References

1. Adler, I.: The equivalence of linear programs and zero-sum games. International Journal of Game Theory **42**, 165–177, Springer (2013)
2. Astorino, A., Fuduli, A.: Spherical separation with infinitely far center. Soft Computing **24(23)**, 17751–17759 (2020)
3. Bazaraa, M. S., Jarvis, J. J., Sherali, H. D.: Linear Programming and Network Flows. John Wiley & Sons (1990)
4. Bosch, S., Güntzer, U., Remmert, R.: Non-Archimedean Analysis: A Systematic Approach to Rigid Analytic Geometry. Springer Berlin Heidelberg (2012)
5. Cavoretto, R., De Rossi, A., Mukhametzhanov, M. S., Sergeyev, Y. D.: On the search of the shape parameter in radial basis functions using univariate global optimization methods. Journal of Global Optimization **79(2)**, 305-327 (2021)
6. Cococcioni, M., Pappalardo, M., Sergeyev, Y. D.: Lexicographic multi-objective linear programming using grossone methodology: Theory and algorithm. Applied Mathematics and Computation **318**, 298–311 (2018)
7. Cococcioni, M., Cudazzo, A., Pappalardo, M., Sergeyev, Y. D.: Grossone Methodology for Lexicographic Mixed-Integer Linear Programming Problems. In: Sergeyev, Y. D., Kvasov D. E. (eds) Numerical Computations: Theory and Algorithms. NUMTA 2019. Springer Lecture Notes in Computer Science **11974**, 337–345 (2020)
8. Cococcioni, M., Cudazzo, A., Pappalardo, M., Sergeyev, Y. D.: Solving the Lexicographic Multi-Objective Mixed-Integer Linear Programming Problem Using Branch-and-Bound and Grossone Methodology. Communications in Nonlinear Science and Numerical Simulation **84**, 105177 (2020)
9. Cococcioni, M., Fiaschi, L.: The Big-M method with the numerical infinite M. Optimization Letters, 1–14 (2020)
10. Cococcioni, M., Fiaschi, L., Lambertini, L.: Non-Archimedean zero-sum games. Journal of Computational and Applied Mathematics **393**, 113483 (2021)
11. D'Alotto, L.: Infinite Games on Finite Graphs Using Grossone. In: Sergeyev, Y. D., Kvasov D. E. (eds) Numerical Computations: Theory and Algorithms. NUMTA 2019. Springer Lecture Notes in Computer Science **11974**, 346–353 (2020)
12. D'Alotto L.: Infinite games on finite graphs using grossone. Soft Computing **24** 17509-17515 (2020)
13. Dantzig, G. B.: Programming in a linear structure. United Air Force, Washington, D.C. (1948)

14. Dantzig, G. B., Thapa, M. N.: Linear Programming 1: Introduction. Springer-Verlag, New York (1997)
15. Dantzig, G. B., Thapa, M. N.: Linear Programming 2: Theory and Extensions. Springer-Verlag, New York (2003)
16. De Cosmis, S., De Leone, R.: The use of Grossone in Mathematical Programming and Operations Research. Applied Mathematics and Computation **218**, 8029–8038 (2021)
17. De Leone, R., Egidi, N., Fatone, L..: The use of grossone in elastic net regularization and sparse support vector machines. Soft Computing **24(23)**, 17669–917677 (2020)
18. De Leone, R., Fasano, G., Roma, M., Sergeyev, Y. D.: Iterative grossone-based computation of negative curvature directions in large-scale optimization. Journal of Optimization Theory and Applications **186(2)**, 554-589 (2020)
19. De Leone, R., Fasano, G., and Sergeyev, Y. D.: Planar methods and grossone for the conjugate gradient breakdown in nonlinear programming. Computational Optimization and Applications **71(1)**, 73-93 (2018)
20. Downs, A.: An Economic Theory of Democracy. Harper and Row (1957)
21. Falcone, A., Garro, A., Mukhametzhanov, M. S. S., Sergeyev, Y. D.: Representation of grossone-based arithmetic in simulink for scientific computing. Soft Computing **24**, 17525–17539 (2020)
22. Falcone, A., Garro, A., Mukhametzhanov, M. S., Sergeyev, Y. D.: A Simulink-based software solution using the Infinity Computer methodology for higher order differentiation. Applied Mathematics and Computation **409**, 125606 (2021)
23. Fiaschi, L., Cococcioni, M.: Numerical Asymptotic Results in Game Theory Using Sergeyev's Arithmetic of Infinity. International Journal on Unconventional Computing **14**, 1–25 (2018)
24. Fiaschi, L., Cococcioni, M.: Generalizing Pure and Impure Iterated Prisoner's Dilemmas to the Case of Infinite and Infinitesimal Quantities. In: Sergeyev, Y. D., Kvasov D. E. (eds) Numerical Computations: Theory and Algorithms. NUMTA 2019. Springer Lecture Notes in Computer Science **11974**, 370–377 (2020)
25. Fiaschi, L., Cococcioni, M.: Non-Archimedean Game Theory: A Numerical Approach. Applied Mathematics and Computation, 125356 (2020)
26. Gaudioso M., Giallombardo G., Mukhametzhanov M.: Numerical infinitesimals in a variable metric method for convex nonsmooth optimization. Applied Mathematics and Computation **318**, 312–320 (2018).
27. Hotelling, H.: Stability in Competition. The Economic Journal **39**, 41–57 (1929)
28. Kantorovič, L. V.: Mathematical methods of organizing and planning production. Publication House of the Leningrad State University (1939)
29. Levi-Civita, T.: Sugli infiniti ed infinitesimi attuali, quali elementi analitici. Tip. Ferrari (1893)
30. Levi-Civita, T.: Sui numeri transfiniti. Tipografia della R. Accademia dei Lincei (1898)
31. Myerson, R. B.: Game Theory. Harvard university press (2013)
32. Nikaidô, H., Isoda, K.: Note on non-cooperative convex games. Pacific Journal of Mathematics **5**, 807–815 (1955)
33. Owen, G.: Game Theory. Academic Press (1995)
34. Peldschus, F.: Experience of the game theory application in construction management. Technological and Economic Development of Economy **14**, 531–545 (2008)
35. Petrosyan, L. A., Zenkevich, N. A.: Game Theory. World Scientific Publishing Co. Pte. Ltd., 9824 (2016)
36. Radner, R.: Collusive behavior in noncooperative epsilon-equilibria of oligopolies with long but finite lives. Journal of Economic Theory **22**, 136–154 (1980)
37. Rizza, D.: Numerical methods for infinite decision-making processes. International Journal of Unconventional Computing **14**, 139–158 (2019)
38. Robinson, A.: Non-standard analysis. Princeton University Press (2016)
39. Schneider, P.: Nonarchimedean functional analysis. Springer Science & Business Media (2013)
40. Sergeyev, Y. D.: Computer system for storing infinite, infinitesimal, and finite quantities and executing arithmetical operations with them. USA patent 7,860,914 (2010)
41. Sergeyev, Y. D.: The Olympic Medals Ranks, lexicographic ordering and numerical infinities. Math. Intelligencer **37**, 4–8 (2015)

42. Sergeyev, Y. D.: Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems. EMS Surveys in Mathematical Sciences **4**, 219–320 (2017)
43. Sergeyev, Y. D.: Independence of the Grossone-Based Infinity Methodology from Non-standard Analysis and Comments upon Logical Fallacies in Some Texts Asserting the Opposite. Foundations of Science **24**, 153–170 (2019)
44. Sergeyev, Y. D., Nasso M. C., Mukhametzhanov, M. S. and Kvasov, D.: Novel local tuning techniques for speeding up one-dimensional algorithms in expensive global optimization using Lipschitz derivatives. Journal of Computational and Applied Mathematics **383**, 113134 (2021)
45. Shapley, L. S., Snow, R. N.: Basic Solutions of Discrete Games. Annals of Mathematics Studies (1950)
46. Soleimani-Damaneh, M.: Modified Big-M method to recognize the infeasibility of linear programming models. Knowledge-Based Systems **21**, 377–382 (2008)
47. Vanderbei, R. J. et al.: Linear programming. Springer (2015)
48. Washburn, A. R.: Two-person zero-sum games. Springer (2014)