

Article

A Non-Archimedean Interior Point Method and Its Application to the Lexicographic Multi-Objective Quadratic Programming

Lorenzo Fiaschi *  and Marco Cococcioni 

Department of Information Engineering, University of Pisa, 56122 Pisa, Italy

* Correspondence: lorenzo.fiaschi@phd.unipi.it

Abstract: This work presents a generalized implementation of the infeasible primal-dual interior point method (IPM) achieved by the use of non-Archimedean values, i.e., infinite and infinitesimal numbers. The extended version, called here the non-Archimedean IPM (NA-IPM), is proved to converge in polynomial time to a global optimum and to be able to manage infeasibility and unboundedness transparently, i.e., without considering them as corner cases: by means of a mild embedding (addition of two variables and one constraint), the NA-IPM implicitly and transparently manages their possible presence. Moreover, the new algorithm is able to solve a wider variety of linear and quadratic optimization problems than its standard counterpart. Among them, the lexicographic multi-objective one deserves particular attention, since the NA-IPM overcomes the issues that standard techniques (such as scalarization or preemptive approach) have. To support the theoretical properties of the NA-IPM, the manuscript also shows four linear and quadratic non-Archimedean programming test cases where the effectiveness of the algorithm is verified. This also stresses that the NA-IPM is not just a mere symbolic or theoretical algorithm but actually a concrete numerical tool, paving the way for its use in real-world problems in the near future.

Keywords: quadratic programming; interior point methods; lexicographic multi-objective optimization; non-standard analysis; infinite/infinitesimal numbers; non-Archimedean scientific computing; fixed-length representations

MSC: 90C51; 90C20; 90C29



Citation: Fiaschi, L.; Cococcioni, M. A Non-Archimedean Interior Point Method and Its Application to the Lexicographic Multi-Objective Quadratic Programming. *Mathematics* **2022**, *10*, 4536. <https://doi.org/10.3390/math10234536>

Academic Editor: Daniel Solow

Received: 24 October 2022

Accepted: 26 November 2022

Published: 30 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Convex quadratic programming (QP) is a widely and deeply studied research topic [1], with plenty of pervasive real-world applications [2]. In spite of the availability of a huge amount of solving algorithms, one of them manifests a pleasant cross-model efficacy at the price of minimal changes in the implementation: the interior point method (IPM) and its variants [3]. Actually, it positively tackles a wide range of optimization tasks, which range from linear programming to cone programming, passing through convex quadratic programming and semidefinite programming.

Another important and of optimization that is growing in interest type is lexicographic multi-objective (linear or nonlinear) programming, which is the task of seeking optimal values of a set of functions (potentially constrained on the domain space) ranked lexicographically (the first objective has priority over the second, the second over the third, and so on [4–8]). Common ways to tackle such a problem are scalarization [9] and preemptive optimization [10] (Chapter 17.7). The scalarization approach transforms the multi-objective problem into a single objective one by computing a weighted sum of the functions according to their priority (the most important one takes a higher weight, the lowest important a lower weight). Unfortunately, the correct a priori choice of the weights is unknown and depends on the function's range and on the optimal solution as well. The lack of any guarantee of correct convergence is not the only drawback, the choice of too different, i.e., very high and very small weights, may induce a loss of precision and numerical instabilities. On the contrary, a preemptive approach optimizes one function at a time and

uses its optimal value as an additional constraint for subsequent tasks. Even if this idea guarantees convergence to optimality, it is not drawbacks-free either. Indeed, it requires solving multiple optimization tasks (one for each function) with increasing complexity (the addition of one constraint after each optimization is completed), which implies wasting computational resources. Furthermore, the nature of the problem can vary and become more complicated over time. For instance, in a QP problem, all the optimizations after the first one shall involve a convex constraint, turning them into quadratically constrained programming problems. As a result, the optimization needs a different, more complex, and less performing algorithm than the one it was supposed to be used at the beginning.

The recent literature has started to reconsider lexicographic problems, trying to solve the issues coming from both scalarization and preemptive approaches leveraging recent developments in numerical non-Archimedean computations [11–13]. Some results involve game theory [14,15], evolutionary optimization [16–18], and linear programming tackled by Simplex-like methods [4,19,20]. Actually, the key idea is to scalarize the problem by adopting infinite and infinitesimal weights. Such a choice of weights guarantees the satisfaction of lexicographic ordering during the optimization of the single-objective constrained (and non-Archimedean) function. In a nutshell, this idea mixes the good properties of preemptive approaches (guarantee of lexicographic ordering satisfaction and convergence to the problem optimum, if any) and scalarization (transformation of the problem into a scalar one, solving it leveraging common and well-established techniques). Thus, the idea of this work is close to the one presented in [4], but this time using an IPM instead of a Simplex algorithm. Once implemented, the new algorithm, called the NA-IPM (non-Archimedean interior point method) will be able to solve not only lexicographic multi-objective linear programming problems but also quadratic ones, and in a polynomial time. Furthermore, by adding only two variables and one non-Archimedean constraint to the original problem, the NA-IPM is able to handle infeasibility and unboundedness transparently, i.e., without the need to cope with them as corner cases within the implementation. This means that ad hoc routines (such as duality gap norm divergence) or more complex embeddings (e.g., homogeneous self-dual model) can be avoided, preventing possible slowing of computations and the need for different solving algorithms. Lexicographic interpretation apart, the NA-IPM is also able to tackle a huge variety of linear and quadratic non-Archimedean optimization problems as well, as opposed to the non-Archimedean extension of the Simplex algorithm in [4]. Examples of such problems are the non-Archimedean zero-sum games presented in [14].

In line with the recent literature, this work refers to those algorithms (as well as models) as fully implemented (described), leveraging standard analysis with the adjective “standard”. This helps to better distinguish them from their non-Archimedean counterparts which, in the majority of the cases, are built over non-standard models (from here the dichotomy). Actually, this is the case of the NA-IPM, which leverages the non-standard model known as Alpha Theory. Equivalently, one could have used the word “Archimedean” in place of “standard”, but it is far less common and it may have led to misunderstandings. It is also worth stressing that the use of a standard or non-standard approach to model a problem does not imply that the problem itself is inherently standard or non-standard. Rather, the theory within which a problem is modeled just defines the perimeter of tools one can use to tackle it and set a limit to the efficacy of the chosen approach as a consequence. This work proposes a non-standard theory to model some classes of optimization problems and proposes the NA-IPM as a solving approach, believing that there are scenarios where its efficacy is higher than the one of standard techniques.

The remainder of the work is structured as follows: Sections 2 and 3 provide the basic knowledge to understand this work, the first reviews IPM and the theory behind it, while the second introduces the non-Archimedean model adopted (Benci’s Alpha Theory); Section 4 presents the NA-IPM, discussing algorithm’s theoretical properties (convergence, complexity, etc.), implementation issues, and handling of infeasibility; Section 5 shows four numerical experiments which highlight the NA-IPM’s effectiveness in linear,

quadratic, feasible, and infeasible problems. Finally, Section 6 concludes the work with few summarizing lines.

2. Reviewing the Standard Interior Point Method for Quadratic Programming

Quadratic programming (QP) [21] is the task of solving a problem of the form of Equation (1) or Equation (2)

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned} \quad (1)$$

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^n$ is the unknown, $Q \in \mathbb{R}^{n \times n}$ is symmetric, $c \in \mathbb{R}^n$ together with Q forms the cost function, while $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are the feasibility constraints, $m, n \in \mathbb{N}$. Whenever Q is also positive semidefinite ($Q \succeq 0$), the QP problem can be solved with polynomial complexity [22]; this will be the case study from now on. A very famous algorithm to solve QP problems is the so-called interior point method (IPM) [3], in any of its many fashions: primal/primal-dual, feasible/infeasible, predictor–corrector, or not. The success of IPM has been so wide and great that it pushed [23] to stress its spirit of unification, which has brought together areas of optimization firmly treated as disjoint. Among the reasons for its practical success, one is its astonishing efficiency for very large-scale linear programming problems (where Simplex-like approaches generally get stuck) [24], the capability to exploit any block-matrix structure in the linear algebra operations [25], as well as the possibility to easily and massively parallelize its computations achieving a significant increase in speed (again, the same does not hold for Simplex-like approaches) [25]. Empirical observations add one more positive feature: even if the best implementation of IPM is proved to find an ε -accurate solution in $\mathcal{O}(\sqrt{n} |\log \varepsilon|)$ iterations, in practice it performs much better than that, converging in almost a constant number of steps regardless the problem dimension n [26] (ε is the optimality tolerance).

This work focuses on the predictor–corrector infeasible primal-dual IPM [27], which is broadly accepted today to be the most efficient one [3]. The next subsection is devoted to resuming the algorithm in question. The reader already familiar with IPM might find it overly lengthy, but it is better to recall here the details of IPM standard implementation in order to more easily introduce its non-Archimedean extension in Section 4.

Description of the Standard IPM Algorithm

First of all, the assumption that the problem is formulated according to (2) holds. If not, one needs to rewrite the problem by adding some slack variables, one for each constraint, without penalizing them in the cost function. Presolving techniques [28] for guaranteeing consistent problem formulations must be applied too. Second, duality theory [29] states that first-order optimality conditions for QP problems (also known as KKT conditions) are the ones reported in Equation (3), where λ are the dual variables and s the dual slack variables (a slight modification which includes an additional parameter μ is provided in Equation (4) and will be discussed later).

$$\begin{cases} Ax = b \\ A^T \lambda + s - Qx = c \\ Xs \mathbf{1} = 0 \\ (x, s) \geq 0 \end{cases} \quad (3)$$

$$\begin{cases} Ax = b \\ A^T \lambda + s - Qx = c \\ Xs \mathbf{1} = \mu \mathbf{1} \\ (x, s) \geq 0 \end{cases} \quad (4)$$

Third, iterative algorithms based on the Newton–Raphson method [30] find the solution of systems as the one in (3) (positivity of x and s excluded) starting from a solution (x^0, λ^0, s^0) and repeatedly solving the linear system in (5) (Equation (6) presents a slight modification with two additional parameters μ and σ which will be introduced in the next lines), where $r_b = Ax - b$ and $r_c = A^T \lambda + s - Qx - c$, while X and S are the diagonal matrices obtained by x and s (the iteration apices are omitted for readability reasons).

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XS\mathbf{1} \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ \sigma\mu\mathbf{1} - XS\mathbf{1} \end{bmatrix} \quad (6)$$

This system is known as Newton’s step equation, and once solved the approximated solutions of (3) are updated accordingly: $(x^{k+1}, \lambda^{k+1}, s^{k+1}) \leftarrow (x^k + \nu\Delta x, \lambda^k + \nu\Delta \lambda, s^k + \nu\Delta s)$ with $\nu \in (0, 1]$ opportunely chosen.

Infeasible primal-dual IPM aims to find optimal primal-dual solutions $(\bar{x}, \bar{\lambda}, \bar{s})$ of (3) by applying a variant of (5) and modifying the search direction as well as the step lengths so that the non-negativity of x and s is strictly satisfied. To do this, the IPM is designed as a path-following algorithm, i.e., an algorithm that tries to stay as close as possible to a specific curve during the optimization [31]. Such a curve, also known as *central path*, is uniquely identified by all the duality-gap values $\mu \in (0, \frac{x^{0T}s^0}{n}]$, where (x^0, λ^0, s^0) is the primal-dual starting point. Its role is to keep the intermediate solutions away from the feasible region’s boundaries as much as possible (from here we get the word “central”). The points belonging to the central path are those satisfying conditions in Equation (4), which is a slight modification of (3), for a given value of μ . The uniqueness of the curve is guaranteed whenever the feasible region admits at least one strictly feasible point, that is a primal-dual solution such that $(x, s) > 0$ (the latter is a very weak assumption). Newton’s step is now computed accordingly with Equation (6), where $\sigma\mu$ is the next duality gap to achieve, $\sigma \in (0, 1)$; furthermore, $\sigma\mu$ uniquely identifies the primal-dual point on the central path to the target. Since Equation (6) approximates (5) more and more closely as μ goes to zero, if the latter converges to anything as $\mu \downarrow 0$ (and it does if the feasible region is bounded), then IPM eventually converges to an optimal solution of the QP problem.

The predictor–corrector version of IPM splits Newton’s direction computation into two stages: predictor and corrector. The predictor one solves the system in (5), i.e., it looks for a solution as optimal as possible, identifying it by the three predictive directions solving Newton’s step equation, namely Δx_p , $\Delta \lambda_p$, and Δs_p . Then, it is the turn of the corrector stage, which operates a restoration of some of the centrality lost during the predictor stage. Indeed, pushing all the way towards the optimum may take the temporary solutions too close to the boundaries, affecting the quality of the convergence due to possible numerical issues. To cope with that, the corrector stage solves the linear system in (6) with the constant term $[0, 0, \sigma\mu\mathbf{1} - \Delta x_p \odot \Delta s_p]$ ($\odot$ indicates the Hadamard product). In most cases, the corrector direction badly (but mildly) affects the optimality of the predictor solution, still guaranteeing an improvement with respect to the previous iterate. Indeed, a good centrality of the temporary solutions is too crucial for a good and fast convergence to be ignored. The final optimizing direction is just the sum of the ones found at the two stages. A final remark: since the arbitrary a priori choice of σ may bias the performance too much, Mehrotra [24] fruitfully proposed an adaptive setting of it to $\left(\frac{\mu^{\text{new}}}{\mu}\right)^3$, where $\mu^{\text{new}} = \frac{(x+\Delta x)^T(s+\Delta s)}{n}$.

There are three aspects left to be uncovered to exhaustively present the IPM algorithm used in this work. They consist of three crucial issues of any iterative algorithm: problem infeasibility, starting point choice, and algorithm termination. Multiple ways to address

infeasibility exist, two of the most common are iterates divergence [32] and problem embedding [33]. The first one leverages some results on convergence [27], which state that in case of infeasibility the residuals cannot decrease under a positive constant, and, therefore, the iterates must diverge. To identify this phenomenon, it is enough to check at each iteration whether the norm of x and s exceeds a certain threshold ω , i.e., $\|(x, s)\| > \omega$. On the contrary, the second one aims to embed the given problem in a larger one for which a feasible solution always exists (called an augmented solution). Once found, it tells the user whether the original problem admits a feasible solution (in which case it shall allow one to retrieve it from the augmented one) or the opposite (also specifying which of the polyhedra is infeasible: the primal, the dual, or both).

The starting point choice is vital for the quality of the solution, or even for the convergence itself. In the literature, there exist plenty of papers proposing novel ways to initialize an IPM. In this work, the approach used is the one by Mehrotra [24], which first seeks the solution of the two least squares problems in Equation (7) and (8) in order to satisfy the equality constraints, then executes some additional operations to guarantee strict positivity and well centering (even at the expenses of feasibility). The complete procedure is reported in Algorithm 1.

$$\begin{array}{ll} \min_x & x^T x \\ \text{s.t.} & Ax = b \end{array} \quad (7)$$

$$\begin{array}{ll} \min_{\lambda, s} & s^T s \\ \text{s.t.} & A^T \lambda + s - Qx = c \end{array} \quad (8)$$

Algorithm 1 Starting point computation for IPM.

```

1: procedure STARTING_POINT( $A, b, c, Q$ )
2:   /* solution of (7) */
3:    $x = A^T(AA^T)^{-1}b$ 
4:   /* solution of (8) */
5:    $\lambda = (AA^T)^{-1}A(c + Qx)$ 
6:    $s = c + Qx - A^T\lambda$ 
7:   /* guarantee of positivity */
8:    $\delta_x = \max(-\frac{3}{2} \min x_i, 0)$ 
9:    $x = x + \delta_x \mathbf{1}$ 
10:   $\delta_s = \max(-\frac{3}{2} \min s_i, 0)$ 
11:   $s = s + \delta_s \mathbf{1}$ 
12:  /* increment in centrality and well-balancing */
13:   $\delta_x = \frac{1}{2} \frac{x^T s}{s^T \mathbf{1}}$ 
14:   $\delta_s = \frac{1}{2} \frac{x^T s}{x^T \mathbf{1}}$ 
15:   $x = x + \delta_x \mathbf{1}$ 
16:   $s = s + \delta_s \mathbf{1}$ 
17:  return  $x, \lambda, s$ 
```

Finally, the algorithm termination is demanded to three aggregated values, ρ_1 , ρ_2 , and ρ_3 . They describe the degree of KKT condition (3) satisfaction achieved by the current iteration. As soon as all three values are under a user-defined tolerance threshold $\varepsilon \in \mathbb{R}^+$, the algorithm is considered converged. Actually, ρ_1 and ρ_2 refer to primal and dual feasibility, respectively, while ρ_3 is in charge of the complementary degree of x and s . Their definition is

$$\rho_1 = \frac{\|Ax - b\|}{1 + \|b\|}, \quad \rho_2 = \frac{\|A^T \lambda + s - Qx - c\|}{1 + \|c\|}, \quad \rho_3 = \frac{\mu}{1 + |\frac{1}{2}x^T Qx + c^T x|}. \quad (9)$$

A resume of the whole procedure just described can be found in Algorithm 2. The next section introduces the non-Archimedean tools exploited to implement the more general interior point algorithm at the core of this manuscript.

Algorithm 2 Predictor–corrector infeasible primal-dual IPM.

```

1: procedure STANDARD_IPM( $A, b, c, Q, \varepsilon, \omega, \text{max\_it}$ )
2:   /* flag of correct termination */
3:   flag = False
4:    $x, \lambda, s$  = starting_point( $A, b, c, Q$ )
5:   for  $i = 1, \dots, \text{max\_it}$  do
6:     /* compute residuals */
7:      $r_b = Ax - b$     $r_c = A^T \lambda + s - Qx - c$     $r_\mu = x \odot s$ 
8:     /* compute centrality,  $n$  is the length of  $x$  */
9:      $\mu = \frac{r_\mu^T \mathbf{1}}{n}$ 
10:    /* compute KKT conditions satisfaction parameters */
11:     $\rho_1 = \frac{\|r_b\|}{1+\|b\|}$     $\rho_2 = \frac{\|r_c\|}{1+\|c\|}$     $\rho_3 = \frac{\mu}{1+|\frac{1}{2}x^T Qx + c^T x|}$ 
12:    if  $\rho_1 \leq \varepsilon$  and  $\rho_2 \leq \varepsilon$  and  $\rho_3 \leq \varepsilon$  then
13:      /* primal-dual feasible optimal solution found */
14:      flag = True
15:      return  $x, \lambda, s, \text{flag}$ 
16:    /* compute predictor directions solving (3) */
17:     $\Delta x_p, \Delta \lambda_p, \Delta s_p$  = predict( $A, b, c, Q, r_b, r_c, r_\mu$ )
18:    /* compute predictor step size */
19:     $v_{pp} = 0.99 \min(\max_{\bar{v}}\{\bar{v} \mid x + \bar{v}\Delta x_p \geq 0\}, 1)$ 
20:     $v_{pd} = 0.99 \min(\max_{\bar{v}}\{\bar{v} \mid s + \bar{v}\Delta s_p \geq 0\}, 1)$ 
21:     $v = \min(v_{pp}, v_{pd})$ 
22:    /* estimate  $\sigma$  */
23:     $\tilde{x} = x + v\Delta x_p$     $\tilde{s} = s + v\Delta s_p$ 
24:     $\mu^{\text{new}} = \frac{\tilde{x}^T \tilde{s}}{n}$ 
25:     $\sigma = (\frac{\mu^{\text{new}}}{\mu})^3$ 
26:    /* compute corrector directions solving the corresponding Newton's system */
27:     $\Delta x_c, \Delta \lambda_c, \Delta s_c$  = corrector( $A, b, c, Q, \sigma\mu\mathbf{1} - \Delta x_p \odot \Delta s_p$ )
28:    /* compute new direction */
29:     $\Delta x = \Delta x_p + \Delta x_c$     $\Delta \lambda = \Delta \lambda_p + \Delta \lambda_c$     $\Delta s = \Delta s_p + \Delta s_c$ 
30:    /* compute step size */
31:     $v_p = 0.99 \min(\max_{\bar{v}}\{\bar{v} \mid x + \bar{v}\Delta x \geq 0\}, 1)$ 
32:     $v_d = 0.99 \min(\max_{\bar{v}}\{\bar{v} \mid s + \bar{v}\Delta s \geq 0\}, 1)$ 
33:     $v = \min(v_p, v_d)$ 
34:    /* compute target primal-dual solution */
35:     $x = x + v\Delta x$     $\lambda = \lambda + v\Delta \lambda$     $s = s + v\Delta s$ 
36:    if  $\|(x, s)\| > \omega$  then
37:      /* Divergence detected */
38:      return  $x, \lambda, s, \text{flag}$ 
39:  return  $x, \lambda, s, \text{flag}$ 

```

3. Alpha Theory

This section aims to introduce the non-Archimedean model adopted to generalize the IPM to the NA-IPM: Benci's Alpha Theory [34]. Its key ingredient is the infinite number α , whose definition can be given in multiple ways, as usual in mathematics. In this work, the axiomatization of Alpha Theory given in [12] is used. The reason for this choice is that it provides a very plain and application-oriented introduction to the topic, which

also has the property to stress the numerical nature and immediate applicability of such a theory to data science problems. Then, the last part of the section is devoted to showing the strict relation between standard lexicographic QP problems and a particular category of non-Archimedean QP ones. This fact suggests a straightforward application of the NA-IPM to lexicographic problems, as testified by the experimental section.

3.1. Background in Brief

The minimal ground for Alpha Theory consists of three axioms, the first of which introduces a wider field than $\mathbb{R}$, namely $\mathbb{E}$, which actually contains it.

Axiom 1. *There exists an ordered field $\mathbb{E} \supset \mathbb{R}$ whose numbers are called Euclidean numbers.*

The second one aims at describing better some of the elements in $\mathbb{E}$. Actually, it states that $\mathbb{E}$ contains numbers that are finite (since it contains $\mathbb{R}$), infinite, and infinitesimal according to the following definition:

Definition 1. *Given $\xi \in \mathbb{E}$, then*

- ξ is infinite $\iff \forall n \in \mathbb{N}, |\xi| > n$
- ξ is finite $\iff \exists n \in \mathbb{N}, \frac{1}{n} < |\xi| < n$
- ξ is infinitesimal $\iff \forall n \in \mathbb{N}, |\xi| < \frac{1}{n}$

To do this, it introduces the previously mentioned value α by means of the numerosity function num , which in some sense can be seen as a particular counting function. For the sake of rigorousness, let $V(\mathbb{N})$ denote the superstructure on $\mathbb{N}$, namely

$$V(\mathbb{N}) = \bigcup_{i=0}^{\infty} V_i(\mathbb{N}),$$

where ($\mathcal{P}$ is the power set)

$$V_0(\mathbb{N}) = \mathbb{N}, \quad V_{i+1}(\mathbb{N}) = V_i(\mathbb{N}) \cup \mathcal{P}(V_i(\mathbb{N}));$$

then set

$$\mathcal{U} := \{X \in V(\mathbb{N}) \mid X \text{ is countable}\}.$$

Axiom 2. *There exists a function $\text{num}, \text{num} : \mathcal{U} \rightarrow \mathbb{E}$ which satisfies*

- if A is finite $\text{num}(A) = |A|$ ($|\cdot|$ denotes the cardinality of a set)
- $\text{num}(A) < \text{num}(B)$ if $A \subset B$
- $\text{num}(A \cup B) = \text{num}(A) + \text{num}(B) - \text{num}(A \cap B)$
- $\text{num}(A \times B) = \text{num}(A) \cdot \text{num}(B)$
- $\alpha = \text{num}(\mathbb{N})$

More precisely, Axiom 2 not only introduces the infinite number α but also states that its manipulation by means of algebraic functions generates other Euclidean numbers. For instance, the following equivalences hold true in $\mathbb{E}$ and each one individuates a specific number:

$$\alpha \cdot (\alpha + 2) = \alpha^2 + 2\alpha \quad 0 < \frac{1}{\alpha} = \alpha^{-1} < \alpha^0 = 1 < \alpha^1 = \alpha < (\alpha + 1)$$

$$\frac{-10.0\alpha^2 + 16.0 + 42.0\eta^2}{5.0\alpha^2 + 7.0} = -2.0 + 6.0\eta^2$$

where there is the convention to indicate α^{-1} with η , i.e., $\eta := \alpha^{-1}$.

The last axiom introduces the notion of α -limit by means of which any function $f: \mathbb{R}^m \rightarrow \mathbb{R}$ can be extended to a function $f^*: \mathbb{E}^m \rightarrow \mathbb{E}$ satisfying all the first-order

properties of the former. This is a crucial tool of any non-standard model since it guarantees the transfer of such properties from f to f^* . Because of this, it is commonly referred to as the transfer principle [35].

Axiom 3. Every sequence $\varphi : \mathbb{N} \rightarrow \mathbb{R}^m$ has a unique α -limit denoted by $\lim_{n \uparrow \alpha} \varphi(n)$ which satisfies the following properties:

1. if $\xi \in \mathbb{E}^m$, then there exists a sequence $\varphi : \mathbb{N} \rightarrow \mathbb{R}^m$ such that

$$\xi = \lim_{n \uparrow \alpha} \varphi(n)$$

2. if $\varphi(n) = n$, then

$$\lim_{n \uparrow \alpha} \varphi(n) = \alpha$$

3. if $\exists n_0 \in \mathbb{N}$ such that $\forall n \geq n_0, \varphi(n) \geq \psi(n)$, then

$$\lim_{n \uparrow \alpha} \varphi(n) \geq \lim_{n \uparrow \alpha} \psi(n)$$

4. any sequence φ, ψ

$$\lim_{n \uparrow \alpha} \varphi(n) + \lim_{n \uparrow \alpha} \psi(n) = \lim_{n \uparrow \alpha} (\varphi(n) + \psi(n)),$$

$$\lim_{n \uparrow \alpha} \varphi(n) \cdot \lim_{n \uparrow \alpha} \psi(n) = \lim_{n \uparrow \alpha} (\varphi(n) \cdot \psi(n)).$$

Leveraging Axiom 3, any real-valued function $f : \mathbb{R}^m \rightarrow \mathbb{R}$ can be extended to a function $f^* : \mathbb{E}^m \rightarrow \mathbb{E}$ by setting [12]

$$f^*(\xi) = f^*\left(\lim_{n \uparrow \alpha} \varphi(n)\right) = \lim_{n \uparrow \alpha} f(\varphi(n)). \quad (10)$$

A similar extension is possible for multi-valued functions and sets as well. In the first case, for any $f : \mathbb{R}^m \rightarrow \mathbb{R}^q$ with $f(x) = \{f_1(x), \dots, f_q(x)\}$ and $f_i : \mathbb{R}^m \rightarrow \mathbb{R} \forall i = 1, \dots, q$, $f^* : \mathbb{E}^m \rightarrow \mathbb{E}^q$ is defined as $f^*(\xi) = \{f_1^*(\xi), \dots, f_q^*(\xi)\}$ where $f_i^* = \lim_{n \uparrow \alpha} f_i \forall i = 1, \dots, q$. In the second case, given a set F , F^* is defined as $F^* = \{\lim_{n \uparrow \alpha} f \mid f \in F\}$.

Notice that the use of the word “extension” is fully justified by the fact that $f^*(x) = f(x) \forall x \in \mathbb{R}$. In the remainder of the work, when no ambiguity is possible, the “*” will be omitted, and, therefore, f and f^* will be denoted by the same symbol.

Another important implication of Axioms 1-3 is that any $\xi \in \mathbb{E} \setminus \{0\}$ admits the following representation [34]:

$$\xi = \sum_{i=1}^{\infty} \psi_i \alpha^{g_{\xi}^{(i)}}, \quad \psi_i \neq 0, i \in \mathbb{N}$$

where $\psi_i \in \mathbb{R}$, $g_{\xi} : \mathbb{N} \rightarrow \mathbb{R}$ is a monotonic decreasing function. Finally, we give some definitions which are useful in the remainder of the work.

Definition 2 (Monosemium). $\xi \in \mathbb{E}$ is called monosemium if and only if $\xi = \psi \alpha^i$, $\psi \in \mathbb{R}$, $i \in \mathbb{R}$.

Definition 3 (Leading monosemium). Given $\xi \in \mathbb{E}$, $\psi_1 \alpha^{g_{\xi}^{(1)}}$ is called leading monosemium.

Definition 4 (Leading monosemium function). The leading monosemium function $\text{lead_mon} : \mathbb{E} \rightarrow \mathbb{E}$ is a non-Archimedean function that maps each Euclidean number in its leading monosemium.

Definition 5 (Order of magnitude). The order of magnitude is a function $\mathcal{O} : \mathbb{E} \setminus \{0\} \rightarrow \mathbb{E} \setminus \{0\}$ such that $\forall \xi \in \mathbb{E} \setminus \{0\} \mathcal{O}(\xi) = \alpha^{g_{\xi}^{(1)}}$.

Definition 6 (Smallest order of magnitude). Let one indicate with $\mathbb{E}_{lim} \subset \mathbb{E}$ the set of Euclidean numbers represented by a limited number of monosemia, i.e., $\mathbb{E}_{lim} := \{\xi \in \mathbb{E} \setminus \{0\} \mid \xi = \sum_{i=1}^l \psi_i \alpha^{\xi(i)}, \psi_i \neq 0, l \in \mathbb{N}\}$. The smallest order of magnitude is a function $o: \mathbb{E}_{lim} \rightarrow \mathbb{E}_{lim}$ such that $\forall \xi \in \mathbb{E}_{lim} o(\xi) = \alpha^{\xi(l)}$.

A common generalization of the last two functions to vectorial spaces is the following:

$$\forall \xi \in \mathbb{E}^n \setminus \{0\} \quad \mathcal{O}(\xi) = \max_{i=1, \dots, n} \mathcal{O}(\xi_i) \quad \forall \xi \in \mathbb{E}_{lim}^n \quad o(\xi) = \min_{i=1, \dots, n} o(\xi_i).$$

Definition 7 (Infinitesimal number). A way to indicate that $\xi \in \mathbb{E}$ is infinitesimal (equivalent to the one in Definition 1) is by the inequality $\mathcal{O}(\xi) < \alpha^0$, which is usually indicated as $\xi \approx 0$.

3.2. How To Handle Lexicographic Optimization

This subsection aims to introduce Proposition 1, which builds a bridge between standard lexicographic QP problems and non-Archimedean quadratic optimization. First, let one formally introduce standard and non-Archimedean optimization models. Then, the result comes straightforwardly.

Definition 8 (Standard optimization problem). An optimization problem of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad i = 1, \dots, n, \\ & h_j(x) = 0 \quad j = 1, \dots, m \end{aligned} \quad (11)$$

is said to be standard (or Archimedean) if and only if f , and all g_i and h_j are real-valued multivariate functions.

Definition 9 (Non-Archimedean optimization problem). An optimization problem of the form (11) is said to be non-Archimedean if and only if at least one of the functions f , g_i , or h_j is non-Archimedean, i.e., at least one of them has values in $\mathbb{E}^k$ (with $k \in \mathbb{N}$) or maps the input into values in $\mathbb{E}$.

In the case of QP problems as (1) and (2), this is equivalent to saying that either $x \in \mathbb{E}^k$ or at least one entry of A , b , c or Q is non-Archimedean.

Proposition 1. Consider a lexicographic optimization problem whose objective functions $f_1, \dots, f_n$ are real functions, and the priority is induced by the natural order. Then, there exists an equivalent scalar program over the same domain, whose objective function is non-standard and has the following form:

$$F(x) = \beta_1 f_1(x) + \dots + \beta_n f_n(x),$$

where $\frac{\beta_{i+1}}{\beta_i} \approx 0, i = 1, \dots, n-1$.

Proof. The proof will show that the two programs share the same global maxima. Very similar considerations hold for local maxima and global and local minima but are omitted for brevity.

Let Ω be the domain of the two programs, and $\omega \in \Omega$ be a global maximum of the lexicographic optimization problem, i.e., $\nexists \omega' \in \Omega$ such that $f_1(\omega') > f_1(\omega)$, or $\nexists i = 2, \dots, n$ such that $f_j(\omega') = f_j(\omega), j = 1, \dots, i-1$, and $f_i(\omega') > f_i(\omega)$. However, this is true if and only if $\nexists \omega' \in \Omega$ such that $F(\omega') > F(\omega)$, by the definition of F itself. The latter fact exactly means that ω is a global maximum for the scalar non-standard program. $\square$

The choice of weights $\{\beta_i\}_{i=1}^n$ adopted in this work is $\beta_i = \alpha^{1-i}, i = 1, \dots, n$. As an example, consider the two-objective lexicographic optimization problem in (12). It seeks for the point in the unitary cube which minimizes the first objective $f_1(x) = x_1^2 + x_2^2$.

In case more than one such point exists, as in this example where the whole segment $\mathcal{S} = \{x \in \mathbb{R}^3 \mid x_1 = x_2 = 0, x_3 \in [0, 1]\}$ optimizes f_1 , then such a set of candidate optimal solutions is refined considering the second objective $f_2(x) = x_3 + x_2$. It is easy to verify that f_2 selects as optimal only one point in $\mathcal{S}$, namely the origin.

According to Proposition 1, a possible non-Archimedean reformulation of (12) is the one in (13). Here, the objective function $\zeta(x)$ is built by summing the first objective weighed by $\beta_1 = 1$ with the second one weighed by $\beta_2 = \eta$, i.e., $\zeta(x) = f_1(x) + f_2(x)\eta$. Notice that the request $\frac{\beta_2}{\beta_1} \approx 0$ holds since $\frac{\beta_2}{\beta_1} = \frac{\eta}{1} = \eta \approx 0$. Furthermore, it is clear why the origin also optimizes the second problem. Indeed, as soon as $x_1 \geq 0$ or $x_2 \geq 0$ then $\zeta(x)$ assumes positive finite values, while if $x_3 \geq 0$ then $\zeta(x)$ assumes positive infinitesimal values. On the other hand, in the origin, it holds $\zeta(x) = 0$, which is a value smaller than any other assumed by the objective function in the previous two cases.

$$\begin{aligned} \text{lex } \min_{x \in \mathbb{R}^3} \quad & x_1^2 + x_2^2, x_3 + x_2 \\ \text{s.t.} \quad & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1, \\ & 0 \leq x_3 \leq 1 \end{aligned} \quad (12)$$

$$\begin{aligned} \min_{x \in \mathbb{R}^3} \quad & x_1^2 + x_2^2 + (x_3 + x_2)\eta \\ \text{s.t.} \quad & 0 \leq x_1 \leq 1, \\ & 0 \leq x_2 \leq 1, \\ & 0 \leq x_3 \leq 1 \end{aligned} \quad (13)$$

4. Non-Archimedean Interior Point Method

This section aims to introduce a non-Archimedean extension of the predictor–corrector primal-dual infeasible IPM described in Section 2, called by the authors the non-Archimedean IPM (NA-IPM). Such an algorithm is able to solve non-Archimedean quadratic optimization problems and, as a corner case, standard QP ones as well. Three immediate and concrete advantages in leveraging it are: (i) lexicographic QP problems can be solved as they were unprioritized ones (thanks to Proposition 1); (ii) the problem of infeasibility and unboundedness disappears; (iii) more general QP optimization problems can be modeled and solved (Alpha Theory helps in modeling problems that are difficult to model without it, sometimes it is even impossible).

The algorithm pseudocode is presented in Algorithm 3. To better understand it, the next three subsections are devoted to discussing some delicate aspects of it and of its implementation: unboundedness and infeasibility, complexity and convergence properties, and numerical issues that arise when moving from algorithmic description to software implementation. Similarly to what happens for standard algorithms, it is important to stress that many of the proofs in this section assume Euclidean numbers to be represented by finite sequences of monosemia. Indeed, even if the reference set $\mathbb{E}$ defined by Axioms 1–3 admits numbers represented by infinite sequences, it would not be reasonable to use them in a machine and to discuss the algorithm’s convergence. The reasons are two: (i) the algorithm should manage and manipulate an infinite amount of data; (ii) the machine is finite and cannot store all that information. Notice that, at this stage, the focus is not on variable-length representations of Euclidean numbers, as they would slow the computations down [14]. Fixed-length representations, as the ones discussed in [11] are, therefore, preferred, because they are easier to implement in hardware (i.e., they are more “hardware friendly”), as recent studies testify [36].

Algorithm 3 Non-Archimedean predictor–corrector infeasible primal-dual IPM.

```

1: procedure NA-IPM( $A, b, c, Q, \varepsilon, \text{max\_it}$ )
2:   /* Notice that the divergence is dealt with the embedding presented in Section 4.1 */
3:   /* Therefore the flag of correct termination and the threshold  $\omega$  are useless here */
4:   /* Notice also that only  $\varepsilon \in \mathbb{R}^+$ , while  $A, b, c$  and  $Q$  are Euclidean matrices and vectors */
5:    $x, \lambda, s = \text{starting\_point}(A, b, c, Q)$ 
6:    $x = \text{lead\_mon}(x)$     $\lambda = \text{lead\_mon}(\lambda)$     $s = \text{lead\_mon}(s)$ 
7:   for  $i = 1, \dots, \text{max\_it}$  do
8:     /* compute residuals */
9:      $r_b = Ax - b$     $r_c = A^T \lambda + s - Qx - c$     $r_\mu = x \odot s$ 
10:    /* compute centrality,  $n$  is the length of  $x$  */
11:     $\mu = \frac{r_\mu^T \mathbf{1}}{n}$ 
12:    /* compute KKT conditions satisfaction parameters */
13:     $\rho_1 = \frac{\|Ax - b\|}{1 \mathcal{O}(b) + \|b\|}$ ,    $\rho_2 = \frac{\|A^T \lambda + s - Qx - c\|}{1 \mathcal{O}(c) + \|c\|}$ ,    $\rho_3 = \frac{\mu}{1 \mathcal{O}(\frac{1}{2} x^T Qx + c^T x) + |\frac{1}{2} x^T Qx + c^T x|}$ 
14:    /* check convergence on all the (meaningful) monosemia of the aggregated values (Section 4.3) */
15:    if  $\text{all\_monosemia}(\rho_1) \leq \varepsilon$  and  $\text{all\_monosemia}(\rho_2) \leq \varepsilon$  and
        $\text{all\_monosemia}(\rho_3) \leq \varepsilon$  then
16:      /* primal-dual feasible optimal solution found */
17:      return  $x, \lambda, s$ 
18:    /* compute predictor directions solving (3) */
19:     $\Delta x_p, \Delta \lambda_p, \Delta s_p = \text{predict}(A, b, c, Q, r_b, r_c, r_\mu)$ 
20:    /* keep only the leading monosemia of the gradients (Section 4.3) */
21:     $\Delta x_p = \text{lead\_mon}(\Delta x_p)$     $\Delta \lambda_p = \text{lead\_mon}(\Delta \lambda_p)$     $\Delta s_p = \text{lead\_mon}(\Delta s_p)$ 
22:    /* compute predictor step size */
23:     $v_{pp} = 0.99 \min(\max_{\bar{v}} \{\text{lead\_mon}(\bar{v}) \mid x + \bar{v} \Delta x_p \geq 0\}, 1)$ 
24:     $v_{pd} = 0.99 \min(\max_{\bar{v}} \{\text{lead\_mon}(\bar{v}) \mid s + \bar{v} \Delta s_p \geq 0\}, 1)$ 
25:     $v = \min(v_{pp}, v_{pd})$ 
26:    /* estimate  $\sigma$  */
27:     $\tilde{x} = x + v \Delta x_p$     $\tilde{s} = s + v \Delta s_p$ 
28:     $\mu^{\text{new}} = \frac{\tilde{x}^T \tilde{s}}{n}$ 
29:     $\sigma = \text{lead\_mon}((\frac{\mu^{\text{new}}}{\mu})^3)$ 
30:    /* compute corrector directions solving the corresponding Newton's system */
31:     $\Delta x_c, \Delta \lambda_c, \Delta s_c = \text{corrector}(A, b, c, Q, \sigma \mu \mathbf{1} - \Delta x_p \odot \Delta s_p)$ 
32:    /* compute new direction */
33:     $\Delta x = \Delta x_p + \Delta x_c$     $\Delta \lambda = \Delta \lambda_p + \Delta \lambda_c$     $\Delta s = \Delta s_p + \Delta s_c$ 
34:    /* keep only the leading monosemia of the gradients (Section 4.3) */
35:     $\Delta x = \text{lead\_mon}(\Delta x)$     $\Delta \lambda = \text{lead\_mon}(\Delta \lambda)$     $\Delta s = \text{lead\_mon}(\Delta s)$ 
36:    /* compute step size */
37:     $v_p = 0.99 \min(\max_{\bar{v}} \{\text{lead\_mon}(\bar{v}) \mid x + \bar{v} \Delta x \geq 0\}, 1)$ 
38:     $v_d = 0.99 \min(\max_{\bar{v}} \{\text{lead\_mon}(\bar{v}) \mid s + \bar{v} \Delta s \geq 0\}, 1)$ 
39:     $v = \min(v_p, v_d)$ 
40:    /* compute target primal-dual solution */
41:     $x = x + v \Delta x$     $\lambda = \lambda + v \Delta \lambda$     $s = s + v \Delta s$ 
42:    /* Add infinitesimal centrality to the close-to-zero entries (Section 4.3) */
43:     $x, s = \text{update\_zero\_entries}(x, s)$ 
44:  return  $x, \lambda, s$ 

```

4.1. Infeasibility and Unboundedness

As stated in Section 2, one can approach the problem of infeasibility and unboundedness in two different ways: divergence detection at run time or problem embedding. While the first keeps the problem complexity fixed but negatively affects the computation because of norm divergence polling, the second wastes resources by optimizing a more complex

problem, which is solved efficiently nevertheless. Therefore, the simpler the embedding is, the lesser it affects the performance.

One very simple embedding, proposed by [37–40], consists of the following mapping:

$$\begin{aligned} A \mapsto \tilde{A} &= \begin{bmatrix} A & b - A\mathbf{1} & \mathbf{0} \\ c^T - \mathbf{1}^T + \mathbf{1}^T Q & 0 & -1 \end{bmatrix}, & Q \mapsto \tilde{Q} &= \begin{bmatrix} Q & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ b \mapsto \tilde{b} &= [b^T \quad -\wp_1]^T, & c \mapsto \tilde{c} &= [c^T \quad \wp_2 \quad 0]^T, \end{aligned} \quad (14)$$

where $\wp_1$ and $\wp_2$ are two positive and sufficiently big constants. This embedding adds two artificial variables (one to the primal and one to the dual problem) and one slack variable (to the primal). The goal of adding the artificial variables is to guarantee the feasibility of their corresponding problem, while on their own dual this is equivalent to adding one bounding hyperplane to prevent any divergence. From duality theory indeed, if the primal problem is infeasible, then the dual is unbounded and vice versa. Geometrically, the hyperplane slope is chosen considering a particular conical combination of the constraints and the constant term vector (the one with all coefficients equal to 1). If there is any polyhedron unboundedness, the conical combination outputs a diverging direction and generates a hyperplane orthogonal to it; otherwise, the addition of such constraints has no effect.

On the other hand, the constraint intercept depends on the penalizing weights $\wp_1$ and $\wp_2$, respectively, for the primal and the dual hyperplane. The larger the weight is, the farther is located the corresponding bound. From the primal perspective instead, $\wp_1$ and $\wp_2$ act as penalizing weights for the artificial variables of the dual and the primal problem, respectively. The need for this penalization comes from the fact that to make the optimization consistent, the algorithm must be driven towards feasible points of the original problem, if any. By construction, the latter always have artificial variables equal to zero, which means one has to penalize them in the cost function as much as possible in order to force them to that value. More formally, it can be proved that for sufficiently large values of $\wp_1$ and $\wp_2$: (i) the enlarged problem is strictly feasible and bounded; (ii) any solution for the larger problem is also optimal for the embedded one if and only if both the artificial variables are zero [38].

Unfortunately, this idea is unsustainable when moving from theory to practice, i.e., to implementation. Indeed, a good estimate of the weights is difficult to determine a priori, and the computational performance is sensitive to their values [41]. Trying to find a solution, Lustig [42] investigated the optimal directions generated by Newton's step equation when $\wp_1$ and $\wp_2$ are driven to ∞ , proposing a weight-free algorithm based on these directions. Later, Lustig et al. [43] showed that directions coincide with those of an infeasible IPM, without solving the unboundedness issue actually. When considering a set of numbers larger than $\mathbb{R}$ as $\mathbb{E}$, however, an approach in the middle between (14) and the one by Lustig is possible. It consists of the use of *infinitely* large penalizing weights, i.e., in a non-Archimedean embedding. This choice has the effect of infinitely penalizing the artificial variables, while from a dual perspective it locates the bounding hyperplanes infinitely far from the origin. For instance, in the case of a standard QP problem, it is enough to set both $\wp_1$ and $\wp_2$ to α , obtaining the following map

$$b \mapsto \tilde{b} = [b^T \quad -\alpha]^T, \quad c \mapsto \tilde{c} = [c^T \quad \alpha \quad 0]^T.$$

The idea to infinitely penalize an artificial variable is not completely new, it has already been successfully used in the I-Big-M method [20], previously proposed by the author of this work, even if in a discrete context rather than in a continuous one.

Nevertheless, there is still a little detail to take care of. Embedding-based approaches leverage the milestone theorem of duality to guarantee optimal the solution's existence and boundedness. A non-Archimedean version of the duality theorem must hold too, otherwise, non-Archimedean embeddings end up being theoretically not well founded. Thanks to the transfer principle, $\mathbb{E}$ is free from any issue of this kind, as stated by the next proposition.

Proposition 2 (Non-Archimedean Duality). *Given an NA-QP maximization problem, suppose that the primal and dual problems are feasible. Then, if the dual problem has a strictly feasible point, the optimal primal solution set is nonempty and bounded. Vice versa is true as well.*

Proof. The theorem is true thanks to the transfer principle which, roughly speaking, transfers the properties of standard quadratic functions to quadratic non-Archimedean ones. $\square$

If a generic non-Archimedean QP problem is considered instead, setting the weights to α may be insufficient to correctly build the embedding. Actually, their proper choice depends on the magnitude of the values constituting the problem. Proposition 3 gives a sufficient estimate of them; before showing it, however, three preliminary results are necessary. Lemmata 1–3 address them. All these three lemmata make use of the functions $\mathcal{O}(\cdot)$ and $o(\cdot)$ provided in Section 3 as Definition 4 and 5, respectively. In particular, Lemma 1 provides an upper bound to the magnitude of the entries of the solutions x of a non-Archimedean linear system $Ax \leq b$. This upper bound is expressed as a function of the magnitude of the entries of both A and b . Furthermore, Lemma 1 considers the case in which the linear system to solve is the dual feasibility constraint of a QP problem, i.e., it has the form $A^T \lambda - Q\bar{x} \leq c$ with $\bar{x}$ satisfying $A\bar{x} \leq b$. Lemmas 2 and 3 generalize Lemma 1 considering corner cases too.

Lemma 1. *Let the set of primal-dual optimal solutions Ω be nonempty and bounded. Additionally, let $b, [c, Q] \neq 0$, A has full row rank, and its entries are represented by at most l monosemia, i.e., $A_{ij} = \sum_{h=1}^l (a_{ij})^h \alpha^{g(h)}$. Then, any $(\bar{x}, \bar{\lambda}, \bar{s}) \in \Omega$ satisfies*

$$\mathcal{O}(\bar{x}) \leq \mathcal{O}(\tilde{x}) = \min_{j \in J} \frac{\mathcal{O}(b_j)}{o(A_j)}, \quad \mathcal{O}(\bar{\lambda}) \leq \mathcal{O}(\tilde{\lambda}) = \min_{i \in I} \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\tilde{x})])}{o([A^i, 1])},$$

where $J = \{j = 1, \dots, m \mid b_j \neq 0\}$ and $I = \{i = 1, \dots, n \mid c_i \neq 0 \vee Q_i \neq 0\}$.

Proof. By hypothesis, $A\bar{x} = b$ and, therefore, $|A\bar{x}| = |b|$ too. Focusing on the j -th constraint, $j \in J$, it holds

$$|b_j| = \left| \sum_{h=1}^l (A_j)^h \bar{x} \alpha^{g(h)} \right| \geq |(A_j)^l \bar{x} \alpha^{g(l)}|, \quad (15)$$

which implies

$$\mathcal{O}(b_j) \geq \mathcal{O}(|(A_j)^l \bar{x} \alpha^{g(l)}|) = o(A_j) \mathcal{O}(\bar{x}) \implies \mathcal{O}(\bar{x}) \leq \frac{\mathcal{O}(b_j)}{o(A_j)} \implies \mathcal{O}(\bar{x}) \leq \min_{j \in J} \frac{\mathcal{O}(b_j)}{o(A_j)}. \quad (16)$$

The proof for the second part of the thesis is very similar:

$$A^T \bar{\lambda} + I \bar{s} = c + Q\bar{x} \implies |A^T \bar{\lambda} + \bar{s}_i| = |c_i + Q_i \bar{x}| \leq |c_i| + |Q_i \bar{x}| \leq |c_i| + |Q_i \xi|,$$

where $\xi \in \mathbb{E}^n$ is such that $|Q_i \bar{x}| \leq |Q_i \xi|$ and has the form $\xi = \xi^0 \mathcal{O}(\tilde{x})$, $\xi^0 \in \mathbb{R}^n$. Now, following the same guidelines used in (15) and (16), one gets

$$\mathcal{O}(\bar{\lambda}) \leq \mathcal{O}([\bar{\lambda}, \bar{s}]) \leq \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\tilde{x})])}{o([A^i, 1])} \quad \forall i \in I \implies \mathcal{O}(\bar{\lambda}) \leq \min_{i \in I} \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\tilde{x})])}{o([A^i, 1])}.$$

$\square$

Equation (15) may seem analytically trivial, but actually, it underlines a subtle property of non-Archimedean linear systems: the solution can have entries infinitely larger than any number involved in the system itself. As an example, the 2-by-2 linear system below

admits the unique solution $\bar{x} = [\alpha^2, \alpha^2]$. However, each value in the system is finite, i.e., the magnitude of each entry of A and b is $\mathcal{O}(\alpha^0) = \mathcal{O}(1)$:

$$\begin{bmatrix} \eta^2 - 1 & 1 \\ 1 & \eta^2 - 1 \end{bmatrix} x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Notice that Lemma 1 works perfectly here. Indeed, $\mathcal{O}(b) = 1$ and $o(A) = \eta^2$ imply $\mathcal{O}(\tilde{x}) = \alpha^2 \geq \mathcal{O}(\bar{x})$.

Lemma 2. Let either the primal problem be unbounded or $\Omega \neq \emptyset$ be unbounded in the primal variable. Let also b, c, Q , and A satisfy the same hypothesis as in Lemma 1. Then,

$$\mathcal{O}(\bar{x}) \leq \mathcal{O}(\tilde{x}) = \min_{j \in J} \frac{\mathcal{O}(b_j)}{o(A_j)}, \quad \mathcal{O}(\bar{\lambda}) \leq \mathcal{O}(\tilde{\lambda}) = \min_{i \in I} \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\tilde{x})])}{o([A^i, (c_i - 1 + Q_i^T \mathbf{1}), 1])},$$

with I and J as in Lemma 1.

Proof. If the primal problem is unbounded, it means that $\exists x \in \mathbb{E}^n$ such that $Ax = b$, and $\forall x$ such that $Ax = b$, $\nexists \lambda \in \mathbb{E}^m$ such that $A^T \lambda \leq c + Qx$. Nevertheless, a relaxed version of Lemma 1 still holds for the primal polyhedron, that is $\exists \bar{x}$ such that $A\bar{x} = b$ and $\mathcal{O}(\bar{x}) \leq \mathcal{O}(\tilde{x}) = \min_{j \in J} \frac{\mathcal{O}(b_j)}{o(A_j)}$ (the request for optimality is missing). According to (14), a feasible bound for the primal polyhedron is $(c^T - \mathbf{1}^T + \mathbf{1}^T Q)x - \zeta = -\wp_2, \zeta \geq 0$, provided a suitable choice of $\wp_2$. Indeed, it can happen that a wrong value for $\wp_2$ turns the unbounded problem into an infeasible one. This aspect shall be discussed in Proposition 3, which specifies $\wp_2$ as a function of $\bar{x}$.

Choice of $\wp_2$ apart, the addition of the bound to the primal polyhedron guarantees that $\exists x' = (x, \zeta) \in \mathbb{E}^{n+1}$ such that it is feasible for the bounded primal problem and $\exists \lambda' \in \mathbb{E}^{m+1}$ such that $A^T \lambda' + (c^T - \mathbf{1}^T + \mathbf{1}^T Q)^T \zeta \leq c + Qx$ (remember that ζ is the dual variable associated with the new constraint of the primal problem and that $\tilde{Q}x' = Qx$). Following the same reasoning used in Lemma 1, one gets the second part of the thesis

$$\mathcal{O}(\bar{\lambda}) \leq \mathcal{O}([\bar{\lambda}', \bar{\zeta}]) \leq \min_{i \in I} \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\tilde{x})])}{o([A^i, (c_i - 1 + Q_i^T \mathbf{1}), 1])}.$$

The case in which $\Omega \neq \emptyset$ and unbounded in the primal variable is very similar. Together with the assumption $b, [c, Q] \neq \mathbf{0}$, it means that there are plenty of (not strictly) feasible primal-dual optimal solutions, but there does not exist any with maximum centrality. This fact negatively affects IPMs, since they move towards maximum centrality solutions. Therefore, an IPM that tries to optimize such a problem will never converge to any point, even if there are a lot of optimal candidates. To avoid this phenomenon, it is enough to bound such a set of solutions with the addition of a further constraint to the primal problem (which has also the effect to guarantee the existence of the strictly feasible solutions missing in the dual polyhedron). As a result, the very same considerations applied for problem unboundedness work in this case as well, leading to exactly the result in the thesis of this lemma. $\square$

Lemma 3. Let either the primal problem be infeasible or $\Omega \neq \emptyset$ be unbounded in the dual variable. Let also b, c, Q , and A satisfy the same hypothesis as in Lemma 1. Then,

$$\mathcal{O}(\bar{x}) \leq \mathcal{O}(\tilde{x}) = \min_{j \in J} \frac{\mathcal{O}(b_j)}{o([A_j, b - A\mathbf{1}])}, \quad \mathcal{O}(\bar{\lambda}) \leq \mathcal{O}(\tilde{\lambda}) = \min_{i \in I} \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\tilde{x})])}{o([A^i, 1])},$$

with I and J as in Lemma 1.

Proof. In this case $\nexists x$ such that $Ax = b$ but $\forall x \in \mathbb{E}^n \exists \lambda \in \mathbb{E}^m$ such that $A^T \lambda \leq c + Qx$. Enlarging the primal problem in accordance to (14), one has that $\exists x' = (\bar{x}, \zeta) \in \mathbb{E}^{n+1}$

($\zeta \geq 0$) such that $A\bar{x} + (b - A\mathbf{1})\zeta = b$. In addition, it holds $\exists \bar{\lambda}$ such that $A^T \bar{\lambda} \leq c + Q\bar{x}$, provided that $(b - A\mathbf{1})^T \bar{\lambda} \leq \wp_1$ for some suitable choice of $\wp_1$ (which shall be discussed in Proposition 3 as well). By analogous reasoning to the ones used in Lemma 1 and 2, the thesis immediately comes.

The case in which $\Omega \neq \emptyset$ and is unbounded on the dual variable works in the same but symmetric way of the complementary scenario discussed in Lemma 2. Because of this, it implies the bounds stated in the thesis, while the proof is omitted for brevity. $\square$

Proposition 3. *Given an NA-QP problem and its embedding as defined in (14), a sufficient estimate of the penalizing weights is*

$$\mathcal{O}(\wp_2) = \mathcal{O}\left(\alpha \min_{j \in J} \frac{\mathcal{O}(b_j)}{o([A_j, b - A\mathbf{1}])}\right), \quad \mathcal{O}(\wp_1) = \mathcal{O}\left(\alpha \min_{i \in I} \frac{\mathcal{O}([c_i, Q_i \mathcal{O}(\wp_2 \eta)])}{o([A^i, (c_i - 1 + Q_i^T \mathbf{1}), 1])}\right),$$

with I and J as in Lemma 1. In case $J = \emptyset$ then $\wp_2 = \alpha$, while $I = \emptyset$ implies $\wp_1 = \alpha$.

Proof. The extension to the quadratic case of Theorem 2.3 in [38] (proof omitted for brevity) gives the following sufficient condition for $\wp_1$ and $\wp_2$, which holds true even in a non-Archimedean context thanks to the transfer principle:

$$\wp_1 > (c^T - \mathbf{1}^T A + \mathbf{1}^T Q)^T \bar{x}, \quad \wp_2 > (b - A\mathbf{1})^T \bar{\lambda}, \quad (17)$$

where $(\bar{x}, \bar{\lambda}, \bar{s})$ is an optimal primal-dual solution of the original problem, if any. A possible way to guarantee the satisfaction of Equation (17) is to choose $\wp_1$ and $\wp_2$ such that their magnitudes are infinitely higher than the right-hand terms of the inequalities. For instance, one may set

$$\wp_1 \geq \alpha(c^T - \mathbf{1}^T A + \mathbf{1}^T Q)^T \bar{x}, \quad \wp_2 \geq \alpha(b - A\mathbf{1})^T \bar{\lambda},$$

or more weakly

$$\mathcal{O}(\wp_1) \geq \mathcal{O}(\alpha(c^T - \mathbf{1}^T A + \mathbf{1}^T Q)^T \bar{x}), \quad \mathcal{O}(\wp_2) \geq \mathcal{O}(\alpha(b - A\mathbf{1})^T \bar{\lambda}).$$

In case Ω is nonempty and bounded, Lemma 1 holds and provides an estimate on the magnitude of both $\bar{x}$ and $\bar{\lambda}$. In case either the primal problem is unbounded or Ω is unbounded in the primal variable, Lemma 2 applies: the optimal solution is handcrafted by bounding the polyhedron, its magnitude is overestimated by $\bar{x}$, and (17) gives a clue for a feasible choice of $\wp_2$. Similar considerations hold for the case of either primal problem infeasibility or Ω unboundedness in the dual variable, where Lemma 3 is used.

Corner cases are the scenarios where either $J = \emptyset$ or $I = \emptyset$. Since $J = \emptyset$ implies $b = \mathbf{0}$, the primal problem is either unbounded or with unique feasible (and optimal) point $x = \mathbf{0}$. In both cases, it is enough to set $\mathcal{O}(\bar{x}) = 1$. Since $\mathbf{0}$ is a feasible solution, in the case of unboundedness it must exist a feasible point with at least one finite entry and no infinite ones because of continuity. In the other scenario, $\mathbf{0}$ is the optimal solution and, therefore, any finite vector is a suitable upper bound for it. Analogous considerations hold for the case $I = \emptyset$, where a sufficient magnitude bound is $\mathcal{O}(\bar{\lambda}) = 1$. $\square$

4.2. Convergence And Complexity

The main theoretical aspects to investigate in an iterative algorithm are convergence and complexity. Notice that in the case of non-Archimedean algorithms, the complexity of elementary operations (such as the sum) assumes their execution on non-Archimedean numbers, rather than on real ones. Since, theoretically, the NA-IPM is just an IPM able to work with numbers in $\mathbb{E}$, one first result on NA-IPM complexity comes straightforwardly thanks to the transfer principle. It is worth stressing that, as usual, Theorem 1 assumes to apply the NA-IPM to an NA-QP problem whose optimal solutions set is non-empty and bounded.

Theorem 1 (NA-IPM convergence). *The NA-IPM algorithm converges in $\mathcal{O}(n^2 |\log \varepsilon|)$, where $n \in \mathbb{N}$ is the primal space dimension and $\varepsilon \in \mathbb{E}^+$ is the optimality relative tolerance.*

Proof. The theorem holds true because of the transfer principle. $\square$

In spite of this result being remarkable, it is of no practical utility. Indeed, the relative tolerance ε may not be a finite value but an infinitesimal one, making the time needed to converge infinite. However, under proper assumptions, finite time convergence can also be guaranteed, as stated by Theorem 2. Before showing it, some preliminary results are needed and are presented as lemmas. In fact, Lemma 4 guarantees optimality improvement iteration by iteration, Lemma 5 provides a preliminary result used by Lemma 6 which proves the algorithm convergence on the leading monosemium.

Lemma 4. *In the NA-IPM, if $\sigma \in \mathbb{R}^+$ then $\exists \tilde{v} \in [0, 1] \subset \mathbb{R}$ such that $\mu^{(k+1)} \leq (1 - 0.1\tilde{v})\mu^{(k)}$ and $\|(r_b^{(k+1)}, r_c^{(k+1)})\| \leq (1 - 0.1\tilde{v})\|(r_b^{(k)}, r_c^{(k)})\|$.*

Proof. Applying the transfer principle to Lemma 6.7 in [27], it holds true that

$$\tilde{v} = \min\left(\frac{n\sigma}{C}, \frac{\sigma(1-\gamma)}{C}, \frac{0.49n}{C}, 1\right), \quad (18)$$

where C is a positive constant at most finite. Equation (18) immediately implies $\tilde{v} \in [0, 1] \subset \mathbb{E}$ and $\mathcal{O}(\tilde{v}) = \mathcal{O}(\sigma)$. The assumption $\sigma \in \mathbb{R}^+$ completes the proof. $\square$

Lemma 5. *Let $d^{(k)}$ be the right-hand term in (6) at the k -th iteration, and d_μ the vector of its last n entries. If the temporary solution $(x^{(k)}, \lambda^{(k)}, s^{(k)}) \in \mathcal{N}_\infty(\gamma, \beta)$ (see Lemma 6 for its definition), then $\|d_\mu\| < n\mu$.*

Proof. By definition, $\|d_\mu\| = \sqrt{\sum_{i=1}^n (\sigma\mu - x_i s_i)^2}$. Focusing on the radicand, one has

$$\begin{aligned} \sum_{i=1}^n (\sigma\mu - x_i s_i)^2 &= \sum_{i=1}^n \sigma^2 \mu^2 - 2\sigma\mu x_i s_i + (x_i s_i)^2 = n\sigma^2 \mu^2 - 2\sigma\mu \sum_{i=1}^n x_i s_i + \sum_{i=1}^n (x_i s_i)^2 = \\ &= n\sigma^2 \mu^2 - 2n\sigma\mu^2 + \sum_{i=1}^n (x_i s_i)^2 < n\sigma^2 \mu^2 - 2n\sigma\mu^2 + n^2 \mu^2, \end{aligned}$$

where the strict inequality comes from the fact that $x_i s_i \geq \gamma\mu$ by hypothesis, which implies $x_i s_i \leq n\mu - \gamma\mu(n-1) < n\mu$. Considering again the square root, the result comes straightforwardly:

$$\|d_\mu\| < \sqrt{n\sigma^2 \mu^2 - 2n\sigma\mu^2 + n^2 \mu^2} = \mu \sqrt{n\sigma^2 - 2n\sigma + n^2} \leq n\mu.$$

$\square$

Lemma 6. *Let $(x^{(0)}, \lambda^{(0)}, s^{(0)})$ be the NA-IPM starting point, and $M^{(0)}\Delta^{(0)} = d^{(0)}$ be the compact form for Newton's step equation (6) at the beginning of the optimization. Let one rewrite the right-hand term $d^{(k)} = (d^{(k)})^0 + (d^{(k)})^1$, where $(d_i^{(k)})^0 = \text{lead_mon}(d_i^{(k)})$. Call d_r the first $n + m$ entries of $(d_i^{(k)})^0$ and d_μ the last n , i.e., $(d_i^{(k)})^0 = (d_r, d_\mu)$. Then, $\exists k \in \mathbb{N}$ such that $|d_{r_i}| \leq \varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$ and $|d_{\mu_i}| < n\varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$ and the Newton–Raphson method reaches that iteration in $\mathcal{O}(n^2 |\log \varepsilon|)$.*

Proof. As usual, the central path neighborhood is

$$\mathcal{N}_{-\infty}(\gamma, \beta) = \left\{ (x, \lambda, s) \in \mathbb{E}^{2n+m} \mid (x, s) > 0, x_i s_i \geq \gamma \mu, \right. \\ \left. \left\| \begin{pmatrix} r_b^{(k)} \\ r_c^{(k)} \end{pmatrix} \right\| \leq \frac{\left\| \begin{pmatrix} r_b^{(0)} \\ r_c^{(0)} \end{pmatrix} \right\|}{\mu^{(0)}} \beta \mu^{(k)}, \beta \geq 1, \beta, \gamma \in \mathbb{R}^+ \right\}.$$

Lemma 4 and μ 's positivity implies that $\mathcal{O}(\mu^{(k+1)}) \leq \mathcal{O}(\mu^{(k)}) \forall k \in \mathbb{N}$. The application of the transfer principle to Theorem 6.2 in [27] guarantees that $\exists k' \in \mathbb{N}$ such that $\mu^{(k')} \leq \varepsilon \mathcal{O}(\mu^{(0)})$ holds true and the Newton–Raphson algorithm reaches that iteration in $\mathcal{O}(n^2 |\log \varepsilon|)$. Together, Lemma 4 and $(x^{(k)}, \lambda^{(k)}, s^{(k)}) \in \mathcal{N}_{-\infty}(\gamma, \beta)$ guarantee that $\exists k''$ (reached in polynomial time as well) such that $\|(r_b^{(k'')}, r_c^{(k'')})\| \leq \varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$ too. Set $k = \max(k', k'')$. Then, one has $\mu^{(k)} \leq \varepsilon \mathcal{O}(\mu^{(0)}) \implies \text{lead_mon}(\mu^{(k)}) \leq \varepsilon \mathcal{O}(\mu^{(0)})$ and $\|(r_b^{(k)}, r_c^{(k)})\| \leq \varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|) \implies \text{lead_mon}(\|(r_b^{(k)}, r_c^{(k)})\|) \leq \varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$. Moreover, by construction it holds $\|d_r^{(k)}\| \leq \text{lead_mon}(\|(r_b^{(k)}, r_c^{(k)})\|)$ and $\|d_\mu^{(k)}\| < n \text{lead_mon}(\mu^{(k)})$ (the latter comes from Lemma 5). Therefore, the following two chains of inequalities hold true:

$$|d_{r_i}^{(k)}| \leq \|d_r^{(k)}\| \leq \text{lead_mon}(\|(r_b^{(k)}, r_c^{(k)})\|) \leq \varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|), \\ |d_{\mu_i}^{(k)}| \leq \|d_\mu^{(k)}\| < n \text{lead_mon}(\mu^{(k)}) \leq n \varepsilon \mathcal{O}(\mu^{(0)}),$$

as stated in the thesis. $\square$

Corollary 1. Let k satisfy Lemma 6, then either

$$x_i < \sqrt{n\varepsilon} \mathcal{O}(x_i) \quad \vee \quad s_i < \sqrt{n\varepsilon} \mathcal{O}(s_i) \quad \forall i = 1, \dots, n.$$

Proof. The result comes straightforwardly from three facts: (i) $x_i s_i < n\mu \forall i = 1, \dots, n$; (ii) $\mu \leq \varepsilon \mathcal{O}(\mu^{(0)})$; (iii) the leading term of entry of x , s , and λ is never zeroed since the full optimizing step is never taken (see lines 19–20 and 31–32 in Algorithm 2). $\square$

We are now ready to provide the convergence theorem for the NA-IPM.

Theorem 2 (NA-IPM convergence). *The NA-IPM converges to the solution of an NA-QP problem in $\mathcal{O}(ln^2 |\log \varepsilon|)$, where $n \in \mathbb{N}$ is the primal space dimension, $\varepsilon \in \mathbb{R}^+$ is the relative tolerance, and $l \in \mathbb{N}$ is the number of consecutive monosemia used in the problem optimization.*

Proof. For the sake of simplicity, assume to represent all the Euclidean numbers in the NA-QP problem by means of the same function of powers $g: \mathbb{N} \rightarrow \mathbb{Q}$. From the approximation up to l consecutive monosemia, one can rewrite $\|(r_b^{(k)}, r_c^{(k)})\| = \sum_{i=1}^l r_i^{(k)} \alpha^{g(i)}$ and $\mu^{(k)} = \sum_{i=1}^l \mu_i^{(k)} \alpha^{g(i)}$. Lemma 6 guarantees that $\exists k$ for which $(d^{(k)})^0$ is ε -satisfied. Now, update the temporary solution substituting each entry of x and s which satisfies Corollary 1 with any feasible value one order of magnitude smaller, e.g., x_i satisfying Corollary 1 is replaced with a positive value of the order $\alpha^{g(j+1)}$, where $j \in \mathbb{N}$ is such that $\mathcal{O}(x_i) = \alpha^{g(j)}$. Actually, they are those variables that are not active at the optimal solution, at least considering the zeroing of $(d^{(k)})^0$ only. Then, recompute $d^{(k)}$, which by construction satisfies $\|d_\mu^{(k)}\| = 0$, $\|d_r^{(k)}\| < \varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$. All these operations have polynomial complexity and do not affect the overall result. Updating the right-hand term as $d^{(k)} \leftarrow d^{(k)} - (d^{(k)})^0$ and zeroing the leading term of those entries whose magnitude is still $\mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$. Next, algorithm iterations are forced to consider the previous $(d^{(k)})^0$ as already fully satisfied. What is actually happening is that the problem now tolerates an infeasibility error whose norm is equal to $\varepsilon \mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|)$. Therefore, one can apply Lemma 6 again to obtain one solution

which is ε -optimal on the second monosemia of $(d^{(0)})^0$ too, and this result is achieved with a finite number of iterations and in polynomial complexity. Repeating the update-optimization procedure for all the l monosemia by means of which $(d^{(0)})^0$ is represented, one obtains one ε -optimal solution on all of them. Since each of the l ε -satisfactions is achieved in $\mathcal{O}(n^2 |\log \varepsilon|)$, then the whole algorithm converges in $\mathcal{O}(ln^2 |\log \varepsilon|)$. $\square$

The next proposition highlights a particular property of the NA-IPM when solving lexicographic QP problems. Actually, it happens that every time μ decreases by one order of magnitude, then one objective is ε -optimized.

Proposition 4. *Consider an NA-QP problem generated from a standard lexicographic one in accordance with Theorem 1 and $\beta_i = \alpha^{1-i} \forall i = 1, \dots, l$. Then, each of the l objectives is ε -optimized in polynomial time and when the i -th one is ε -optimized the magnitude of μ decreases from $\mathcal{O}(\alpha^{1-i})$ to $\mathcal{O}(\alpha^{-i})$ in the next iteration.*

Proof. Assume to start the algorithm with a sufficiently good and well-centered solution, as the one produced by Algorithm 1, then, $\mathcal{O}(\|(r_b^{(0)}, r_c^{(0)})\|) = \mathcal{O}(\mu^{(0)}) = \mathcal{O}(\alpha^0)$. Since $x^{(k)} \in \mathbb{R}^+$ by construction, one can interpret each monosemia in $r_c^{(k)}$ as the satisfaction of the corresponding objective function at the k -th iteration, that is if $r_c^{(k)} = \sum_{i=1}^l r_{c_i} \alpha^{1-i}$ then the first objective lacks r_{c_1} to be fully optimized, the second one lacks r_{c_2} , and so on. Because of Lemma 6, in polynomial time, $(d_i)^0$ is ε -optimized, that is, the KKT conditions (3) are ε -satisfied. In fact, this means that primal-dual feasibility is close to finite satisfaction and centrality is finitely close to zero. There is a further interpretation nevertheless. Interpreting the KKT conditions from a primal perspective their ε -satisfaction testifies that: (i) the primal solution is feasible (indeed, the primal is a standard polyhedron and, therefore, is enough to consider the leading terms of x only, getting rid of the infinitesimal infeasibility r_{b_2}); (ii) the objective function is finitely ε -optimized (which means that the first objective is ε -optimized since the original problem was a lexicographic one and the high-priority objective is the only one associated with finite values of the non-Archimedean objective function); (iii) the approximated solution is very close to the optimal surface of the first objective, roughly speaking it is ε -finitely close. Moreover, the fact that $\|d_\mu\| = 0$ after the updating procedure used in Theorem 2 implies that the magnitude of μ will be one order of magnitude smaller in the next iteration, i.e., it will decrease from $\mathcal{O}(\alpha^0)$ to $\mathcal{O}(\alpha^{-1})$. Since what was just said holds for all the l monosemia (read priority levels of, i.e., objectives in the lexicographic cost function), the proposition is proved true. $\square$

4.3. Numerical Considerations and Implementation Issues

The whole field $\mathbb{E}$ cannot be used in practice since it is too big to fit in a machine. However, the algorithmic field $\hat{\mathbb{E}}$ presented below is enough to represent and solve many real-world problems:

$$\hat{\mathbb{E}} = \left\{ \zeta \in \mathbb{E} \left| \zeta = \sum_{i=1}^l \psi_i \alpha^{g(i)}, l \in \mathbb{N}, \psi_i \in \mathbb{R} \right. \right\} \cup \{0\},$$

where $g : \mathbb{N} \rightarrow \mathbb{Z}$ is a monotone decreasing function and the term “algorithmic field” refers to finite approximations of theoretical fields realized by computers [12]. Similarly to IEEE-754 floating point numbers, which is the standard encoding for real numbers within a machine, a finite dimension encoding for Euclidean numbers in $\hat{\mathbb{E}}$ is needed. In [11,12], the bounded algorithmic number (BAN) representation is presented as a sufficiently flexible and informative encoding to cope with this task. The BAN format is a fixed-length approximation of a Euclidean number. An example of BAN is $\alpha^{-1}(2.4 + 3.9\eta - 2.89\eta^2)$, where the “precision” in this context is given by the degree of the polynomial in η plus 1 (three in this case). The BAN encoding with degree three is indicated as BAN3.

The second detail to take care of when attempting to do numerical computations with Euclidean numbers is the effect of lower-magnitude monosemia on them. For instance, consider a two-objective lexicographic QP whose first objective is degenerate with respect to some entries of x . When solving the problem by means of the NA-IPM, the following phenomenon (which can also be proved theoretically) occurs: the information of the optimizing direction for the secondary objective is stored as an infinitesimal gradient in the solution of Newton's step Equation (6). As an example, assume that $x \in \mathbb{R}^3$ and the entries x_2 and x_3 are degenerate with respect to the first objective. Then, at each iteration the infinitesimal monosemium in the optimizing direction of x_1 assumes a negligible value, while for x_2 and x_3 this is not true: it is significant and grows exponentially in time. In fact, the infinitesimal gradient represents the optimizing direction which must be followed along the optimal (and degenerate) surface of the first objective in order to also reach optimality for the second one. However, such infinitesimal directions do not significantly contribute to the optimization, since the major role is played by the finite entries of the gradient. Therefore, the effect of this infinitesimal information in the gradient only generates numerical instabilities. As soon as the first objective is ε -optimized, i.e., the first objective surface is reached, the optimizing direction still assumes finite values but this time oriented in order to optimize the second objective keeping the first one fixed, while all the infinitesimal monosemia of the gradient assume negligible values. Roughly speaking, it happens as a sort of "gradient promotion" as a result of the change in the objective to optimize. To cope with the issue of noisy and unstable infinitesimal entries in the gradient, two details need to be implemented: (i) after the computation of the gradients (both the predictor and the corrector step), only the leading term of each entry must be preserved, zeroing the remaining monosemia; (ii) after having computed the starting point according to Algorithm 1, again only the leading term of each entry of x , s , and λ must be preserved. These variations do not affect convergence nor the generality of the discussion since the leading terms of the primal-dual solution are the only ones that impact the zeroing of $(d^{(k)})^0$.

The choice of dealing with only the leading terms of the gradients comes in handy to solve another issue during the computations: a good choice for the value to assign to the zeroed entries of x and s during the updating phase discussed in Theorem 2. Actually, it is enough to add one monosemium whose magnitude is such that the following equality holds true: $\mathcal{O}(x_i s_i) = \mathcal{O}(\mu)\eta$. For instance, assume again a two-objective lexicographic QP scenario after having completed the optimization of the first one. It holds true that $\mathcal{O}(\mu) = \alpha^0$ and either x_i or s_i are smaller than $n\epsilon\alpha^0$, say x_i . Then, the updating phase of Theorem 2 sets x_i to a value having magnitude one order smaller. A reasonable approach is to set x_i equal to a monosemium, say ξ , such that $\mathcal{O}(\xi s_i) = \mathcal{O}(\mu)\eta = \eta$. Since $\mathcal{O}(s_i)$ is finite because of Corollary 1, one has $\mathcal{O}(\xi) = \eta$. The naive choice is $\xi = \eta$, but it may be not the best one. Indeed, this approach does not guarantee the generation of a temporary solution with the highest possible centrality, i.e., $x_i s_i = \mu'$, where μ' is the centrality measure after the update. To do it, the value to opt for must be $\xi_i = \frac{\mu'}{z_i}$, where $z_i = \max(x_i, s_i)$ and μ' is a monosemium one order of magnitude smaller than μ and sufficiently (but arbitrarily) far from zero.

Another numerical issue to care about is the computation accuracy due to the number of monosemia used. As a practical example, consider the task to invert the matrix A reported below. Since its entries have magnitudes from α to η , one may think BAN3 encoding is enough to properly compute an approximation of its inverse, which is reported below as A_3^{-1} . However, the product AA_3^{-1} testifies the presence of an error whose magnitude is η^2 , quite close to $o(A) = \eta$. Depending on the application, such noise could negatively affect further computations and the final result. Therefore, a good practice is to foresee some additional slots in the non-Archimedean number encoding. For instance, adopting the BAN5 standard, the approximated inverse A_5^{-1} manifests an error with magnitude η^4 (see matrix AA_5^{-1}), definitely a safer choice even if at the expense of extra computational effort.

$$\begin{aligned}
A &= \begin{bmatrix} \alpha & -\alpha & 2\eta \\ 2\alpha & \eta & -\alpha \\ \eta & 2\alpha & -\alpha \end{bmatrix}, & A_3^{-1} &= \begin{bmatrix} 0.25\alpha - 0.125\eta & -0.12\alpha + 0.5\eta & 0.12\alpha \\ 0.25\alpha - 0.125\eta & -0.12\alpha & 0.12\alpha + 0.5\eta \\ 0.50\alpha & -0.25\alpha - 0.125\eta & 0.25\alpha + 0.125\eta \end{bmatrix}, \\
A_5^{-1} &= \begin{bmatrix} 0.25\alpha - 0.125\eta + 0.0625\eta^3 & -0.12\alpha + 0.5\eta - 0.0312\eta^3 & 0.12\alpha - 0.219\eta^3 \\ 0.25\alpha - 0.125\eta + 0.0625\eta^3 & -0.12\alpha - 0.281\eta^3 & 0.12\alpha + 0.5\eta + 0.0312\eta^3 \\ 0.5\alpha & -0.25\alpha - 0.125\eta - 0.0625\eta^3 & 0.25\alpha + 0.125\eta + 0.0625\eta^3 \end{bmatrix} \\
AA_3^{-1} &= \begin{bmatrix} 1 & -0.25\eta^2 & 0.25\eta^2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & AA_5^{-1} &= \begin{bmatrix} 1 & -0.12\eta^4 & 0.12\eta^4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Finally, the last detail concerns terminal conditions. In the standard IPM, execution stops when the three convergence measures ρ_1 , ρ_2 , and ρ_3 in (9) are smaller than the threshold ε . However, in a non-Archimedean context optimality means to be ε -optimal on each of the l monosemia in the objective function, i.e., to satisfy the KKT condition (3) on the first l monosemia in b and c and μ . This means that the terminal condition needs to be modified in order to cope with such a convergence definition. The convergence measures need to be redefined as

$$\begin{aligned}
\rho_1 &= \frac{\|Ax - b\|}{\mathcal{O}(b) + \|b\|}, & \rho_2 &= \frac{\|A^T\lambda + s - Qx - c\|}{\mathcal{O}(c) + \|c\|}, \\
\rho_3 &= \frac{\mu}{\mathcal{O}(\frac{1}{2}x^T Qx + c^T x) + |\frac{1}{2}x^T Qx + c^T x|},
\end{aligned} \tag{19}$$

with the convention $\mathcal{O}(0) = \alpha^0 = 1$. In this way, one has the guarantee that ρ_1 , ρ_2 , and ρ_3 are finite values when close to optimality. To better clarify this concept, consider the case in which b has only infinitesimal entries. This implies that the norm of b is infinitesimal too. In case the convergence measures in (9) are used, the denominator of ρ_1 is a finite number. Therefore, any finite approximation of b , i.e., any primal solution x such that $\|Ax - b\|$, is finite induces a finite value for ρ_1 . This is definitely a bad behavior since it is natural to expect that: (i) the convergence measures are finite numbers only if the optimization is close to optimality; (ii) their leading monosemium is smaller than ε when the leading monosemium of the residual norm ($\|Ax - b\|$ in this case) is small as well. However, this is not the case in the current example as ρ_1 assumes finite values for approximation errors of b which are infinitely larger than its norm. Using the definitions in (19) instead, the issue is solved since now finite approximations of b are mapped into infinite values of ρ_1 , while infinitesimal errors are mapped into finite ones. In fact, the introduction of the magnitude of the constant term vectors in the definitions avoids the bias which would have been introduced by an a priori choice of the magnitude of the constant term added to the denominator of the convergence measures.

Then, feasibility on the primal problem is ε -achieved when the absolute value of the first l_b monosemia in ρ_1 are smaller than ε , i.e., assuming $b = \sum_{i=1}^{l_b} b^i \alpha^{g(i)}$ and $\rho_1 = \sum_{i=1}^l \rho_1^i \alpha^{1-i}$, a sufficient level of optimality is reached when $|\rho_1^i| < \varepsilon \forall i = 1, \dots, l_b, l \geq l_b$. Similar considerations hold for c and μ as well.

5. Numerical Experiments

This section presents four problems by means of which the efficacy of the NA-IPM is tested. The first, the third, and the fourth are lexicographic ones, while the second involves non-Archimedean numbers also in the constant term b . They are listed in increasing order of difficulty, from LP to QP problems, from two to three objective functions, passing through unboundedness tackling. In all the experiments, ε is set to 10^{-8} , a de facto standard choice.

5.1. Experiment 1: Two-Objective LP

The first experiment uses a benchmark already exploited to evaluate the efficacy of non-Archimedean optimizers [4,20]. In [4], for instance, the algorithm used was a non-Archimedean version of the Simplex method, able to deal with non-Archimedean cost functions and standard constraints. On the contrary, here the adopted algorithm is a non-Archimedean generalization of the primal-dual IPM, which is intrinsically implemented to cope with non-Archimedean functions both at the level of cost function and constraints. Another difference is the fact that Simplex-like algorithms are discrete ones, while IPMs are continuous procedures.

The problem formulation is in Equation (20). Geometrically, the first and most important objective $c = [8, 12]^T$ identifies as optimal the whole segment linking the point $(0, 70)$ to $(30, 50)$, highlighted in red in Figure 1. When considering also the second objective $d = [14, 10]^T$, only one point of this segment turns out to be truly optimal: the vertex $\xi_2 = (30, 50)$. A red star indicates it in Figure 2. Equation (21) reports the non-Archimedean representation of the problem: the two objectives are scalarized into one single cost function by a weighting sum: the first one is left as is while the second is scaled down by the factor η , testifying to the precedence (read importance) relation between the two (recall from Section 3 that η is an infinitesimal value, defined as the reciprocal of α).

$$\begin{aligned} \text{lex max}_x \quad & 8x_1 + 12x_2, 14x_1 + 10x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 120, \\ & 2x_1 + 3x_2 \leq 210, \\ & 4x_1 + 3x_2 \leq 270, \\ & x_1 + 2x_2 \geq 60, \\ & x_1, x_2 \geq 0 \end{aligned} \quad (20)$$

$$\begin{aligned} \max_x \quad & (8 + 14\eta)x_1 + (12 + 10\eta)x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 120, \\ & 2x_1 + 3x_2 \leq 210, \\ & 4x_1 + 3x_2 \leq 270, \\ & x_1 + 2x_2 \geq 60, \\ & x_1, x_2 \geq 0 \end{aligned} \quad (21)$$

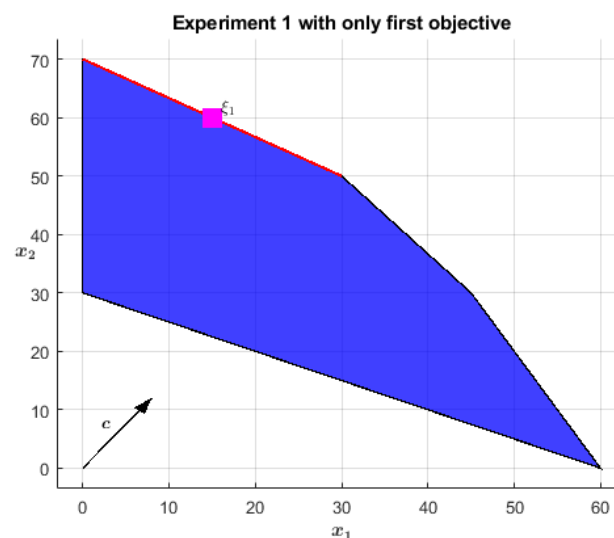


Figure 1. The optimal segment for the primary objective is in red; ξ_1 is the optimal point a standard IPM would approach.

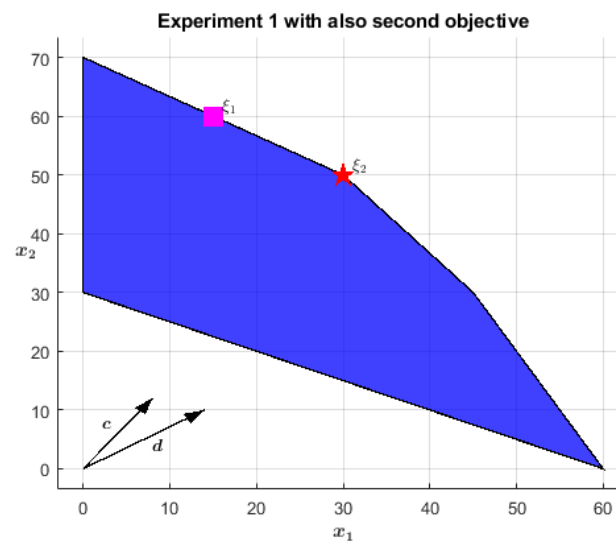


Figure 2. ξ_2 is the only optimal solution when both objectives are considered.

As already stressed, any IPM favors solutions with higher centrality and the NA-IPM is not an exception. This aspect becomes particularly evident in the case of multiple optimal solutions since the optimizing algorithm converges to the analytic center of the optimal surface. Proposition 4 implicitly states that in lexicographic problems the NA-IPM primarily optimizes the first objective, then the second one, and so on. Since in (20) the optimal region for the first objective is a segment, in the first place the NA-IPM moves towards its midpoint $\xi_1 = (15, 60)$, marked by a magenta square in Figure 1. Eventually, the first cost function is ε -optimized and the secondary objective starts to mainly condition the optimization, moving the temporary solution towards ξ_2 until termination conditions are satisfied. This change in leadership is testified when the centrality measure turns from finite to infinitesimal, as discussed in Theorem 2, Proposition 4, and Section 4.3. With reference to Table 1, one can appreciate this phenomenon between lines 5 and 6. At line 5, the current solution is very close to ξ_1 , actually $(18.45, 57.70)$, while μ has assumed for the last time a (very small) finite value. These facts testify that the first objective is ε -satisfied, therefore, in the next step, the algorithm will move towards ξ_2 and show an infinitesimal centrality measure, as confirmed by line 6 where the target point is $(29.88, 50.08)$. Moreover, the improvement in the objective function from step 5 to step 6 is infinitesimal, which further stresses that the algorithm has already reached the first objective ε -satisfaction and it is moving along its degenerate surface. Similar numerical confirmations come from the other experiments and are highlighted in boldface in the associated tables.

Table 1. Iterations of the NA-IPM solving the problem in (20).

Iter	$\mu \in \mathbb{R}$	$x \in \mathbb{R}^2$	$f(x) \in \mathbb{E}$
0	273.00	[98.80 40.51]	−1276.48 − 1790.00 η
1	38.64	[26.94 43.47]	−737.22 − 812.00 η
2	2.97	[18.53 57.56]	−838.97 − 835.00 η
3	0.03	[18.45 57.70]	−839.99 − 835.00 η
4	29.81×10^{-4}	[18.45 57.70]	−840.00 − 835.00 η
5	2.82×10^{-6}	[18.45 57.70]	−840.00 − 835.00 η
6	12.82 η	[29.88 50.08]	−840.00 − 919.00 η
7	0.14 η	[30.00 50.00]	−840.00 − 920.00 η
8	$1.40 \times 10^{-3}\eta$	[30.00 50.00]	−840.00 − 920.00 η
9	$1.41 \times 10^{-5}\eta$	[30.00 50.00]	−840.00 − 920.00 η
10	$4.30 \times 10^{-8}\eta$	[30.00 50.00]	−840.00 − 920.00 η

5.2. Experiment 2: Unbounded Problem

The second experiment aims to numerically show the efficacy of the mild embedding shown in Section 4.1 to cope with infeasibility and unboundedness. As an example, consider the 2D unbounded problem described in Equation (22) and drawn in Figure 3, which is already analytically reported in normal form as in (2) for the sake of clarity. To mitigate the issues coming from the iterates divergence, one can resort to the embedding described in Equation (14), obtaining the strictly feasible and bounded problem in (23). Proposition 3 recommends the use of penalizing weights such that $\mathcal{O}(\varphi_1) = \mathcal{O}(\varphi_2) = \mathcal{O}(\alpha)$; the choice has been $\varphi_1 = \varphi_2 = \alpha$.

$$\begin{aligned}
 & \max_x && x_1 + x_2 \\
 & \text{s.t.} && -2x_1 + x_2 + x_3 = 2 \\
 & && x_1 - 2x_2 + x_4 = 1 \\
 & && x \geq 0 \\
 & && x \in \mathbb{R}^4
 \end{aligned} \tag{22}$$

$$\begin{aligned}
 & \max_x && x_1 + x_2 - \alpha x_5 \\
 & \text{s.t.} && -2x_1 + x_2 + x_3 + 2x_5 = 2 \\
 & && x_1 - 2x_2 + x_4 + x_5 = 1 \\
 & && -x_3 - x_4 - x_6 = -\alpha \\
 & && x \geq 0 \\
 & && x \in \mathbb{E}^6
 \end{aligned} \tag{23}$$

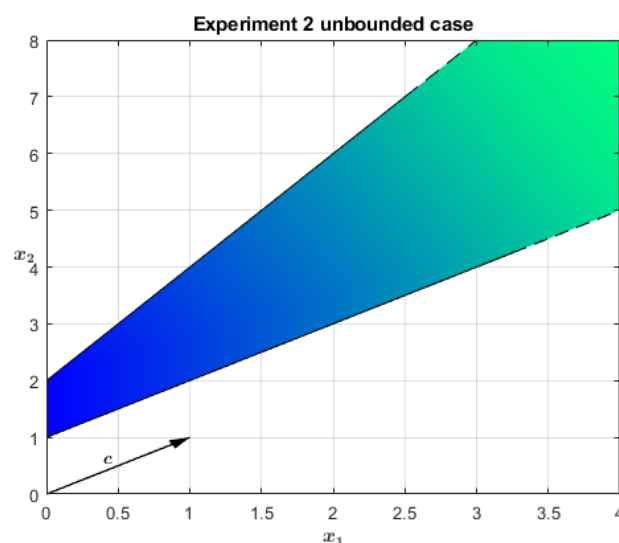


Figure 3. Example of an unbounded primal polyhedron.

Table 2 reports the iterations made by the NA-IPM to solve such an extended problem. As expected, the algorithm converges in a finite number of steps, and the optimal point lies on the bounding hyperplane $-x_3 - x_4 - x_6 = -x_1 - x_2 - 3 - x_6 = -\alpha$ located infinitely far from the origin. Formally, what gives a clue about the unboundedness of the problem is the dual variable λ_3 , see Proposition 3. If the problem is bounded then it must be zero in the optimal solution, while it is equal to 1. In this specific case, however, there is another and more significant indicator: the magnitude of x_1 and x_2 . Since the problem was a standard one before the embedding, if its solution exists it must be finite. In the optimal point found by the NA-IPM instead, x_1 and x_2 are infinite, which tells the user the original problem was unbounded. It may be right to say that, in the current problem, the additional constraint introduced by Equation (14) is equivalent to the constraint $x_1 + x_2 \leq \alpha$ (more precisely to $x_1 + x_2 \leq \alpha - 3$), which would probably have been the first choice of anyone at first looking at Figure 3.

Table 2. Iterations of the NA-IPM solving the problem in (22).

iter	$\mu \in \mathbb{E}$	$x \in \mathbb{E}^2$	$f(x) \in \mathbb{E}$
0	$0.20\alpha^2$	$[0.46\alpha \quad 0.51\alpha]$	$0.25\alpha^2 - 0.96\alpha$
1	$0.03\alpha^2$	$[0.30\alpha \quad 0.32\alpha]$	$1.55 \times 10^{-3}\alpha^2 - 0.62\alpha$
2	$0.02\alpha^2$	$[0.31\alpha \quad 0.32\alpha]$	$2.55 \times 10^{-5}\alpha^2 - 0.64\alpha$
3	$2.03 \times 10^{-5}\alpha^2$	$[0.31\alpha \quad 0.32\alpha]$	$2.55 \times 10^{-7}\alpha^2 - 0.64\alpha$
4	$2.03 \times 10^{-7}\alpha^2$	$[0.31\alpha \quad 0.32\alpha]$	-0.64α
5	$7.40 \times 10^{-10}\alpha^2$	$[0.31\alpha \quad 0.32\alpha]$	-0.64α
6	0.01α	$[0.46\alpha - 1.45 \quad 0.47\alpha - 1.15]$	$-0.92\alpha + 3.28$
7	$2.54 \times 10^{-4}\alpha$	$[0.49\alpha - 1.63 \quad 0.50\alpha - 1.33]$	$-1.00\alpha + 3.01$
8	$2.55 \times 10^{-6}\alpha$	$[0.49\alpha - 1.65 \quad 0.51\alpha - 1.35]$	$-1.00\alpha + 3.00$
9	$2.55 \times 10^{-8}\alpha$	$[0.49\alpha - 1.65 \quad 0.51\alpha - 1.35]$	$-1.00\alpha + 3.00$
10	1.99×10^{-9}	$[0.49\alpha - 1.65 \quad 0.51\alpha - 1.35]$	$-1.00\alpha + 3.00$

The other side of the coin is the problem described in Equation (24) and drawn in Figure 4. In this case, the primal problem is infeasible, which means that now the dual is unbounded. Leveraging Proposition 3 again, the enlarged problem becomes the one in Equation (25). Running the NA-IPM in this extended problem, one appreciates that x_5

is equal to α in the optimal solution, i.e., it is nonzero and the original primal problem is infeasible. As before, another indicator of the primal problem infeasibility is the magnitude of λ_1 and λ_2 , which testifies to the dual problem unboundedness. In particular, they are equal to $\lambda_1 = -0.18\alpha + 0.27$, $\lambda_2 = -0.09\alpha - 0.36$. The algorithm iterations are not reported for brevity.

$$\begin{aligned} \max_x \quad & x_1 + x_2 \\ \text{s.t.} \quad & 2x_1 - x_2 + x_3 = -2 \\ & -x_1 + 2x_2 + x_4 = -1 \\ & x \geq 0 \end{aligned} \quad (24)$$

$$\begin{aligned} \max_x \quad & x_1 + x_2 - \alpha x_5 \\ \text{s.t.} \quad & 2x_1 - x_2 + x_3 - 4x_5 = -2 \\ & -x_1 + 2x_2 + x_4 - 3x_5 = -1 \\ & -x_3 - x_4 - x_6 = -\alpha \\ & x \geq 0 \end{aligned} \quad (25)$$

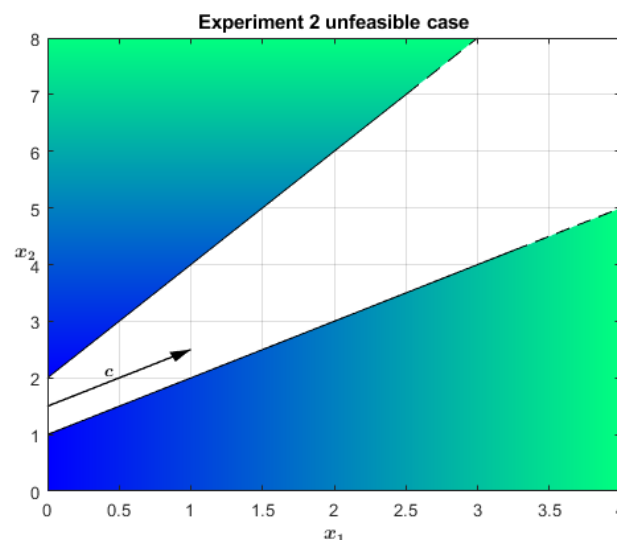


Figure 4. Example of an empty primal polyhedron.

All in all, applying the embedding in (14), the existence of optimal primal-dual solutions is guaranteed and an implementation of the NA-IPM, which does not repeatedly check for infeasibility can be used, helping the performance a lot.

5.3. Experiment 3: Two-Objective QP

The problem faced in this subsection is nonlinear in the first objective and linear in the second one. Its formal construction is reported in Appendix A, while its standard description is in Equation (26) and its non-Archimedean one is in Equation (27):

$$\begin{aligned} \text{lex min}_x \quad & \frac{1}{2}x^T Qx + q^T x, \quad c^T x \\ \text{s.t.} \quad & -x_1 + x_2 + x_3 \leq 1 \\ & -x_1 - x_2 + x_3 \leq 1 \\ & x_1 - x_2 + x_3 \leq 1 \\ & x_1 + x_2 + x_3 \leq 3 \\ & x_3 \geq 0 \end{aligned} \quad (26)$$

$$\begin{aligned}
 \min_x \quad & \frac{1}{2}x^T Q \eta x + (c + q\eta)^T x \\
 \text{s.t.} \quad & -x_1 + x_2 + x_3 \leq 1 \\
 & -x_1 - x_2 + x_3 \leq 1 \\
 & x_1 - x_2 + x_3 \leq 1 \\
 & x_1 + x_2 + x_3 \leq 3 \\
 & x_3 \geq 0
 \end{aligned} \tag{27}$$

$$Q = \begin{bmatrix} 10 & -2 & 4 \\ -2 & 10 & 4 \\ 4 & 4 & 4 \end{bmatrix}, \quad q = \begin{bmatrix} -16 \\ -16 \\ -16 \end{bmatrix}, \quad c = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}.$$

The primal feasible region is a right square pyramid of height one, see Figure 5. Its basis lies on the x_1x_2 plane and has its center at $(1, 1, 0)$ and vertices in $(1, 0, 0)$, $(0, 1, 0)$, $(1, 2, 0)$, and $(2, 1, 0)$. Moving to the cost function, the first objective penalizes points far from the line $\mathcal{L} = (t, t, -2t + 4)$, $t \in \mathbb{R}$, i.e., the axis of the cylinder in Figure 6. The optimal region, highlighted in green in both the figures, is the height of the furthest-from-origin face of the pyramid, identified by the intersection of the latter with the cylinder of radius $\frac{1}{\sqrt{3}}$ and $\mathcal{L}$ as the axis. Finally, the second objective selects from the whole apothem the point as close as possible to the x_1x_2 plane, namely $\zeta_2 = (\frac{1}{2}, \frac{1}{2}, 0)$.

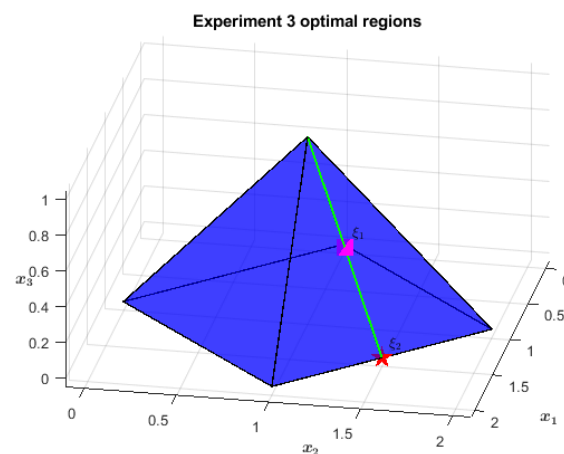


Figure 5. The segment in green is the optimal region for the first objective and ζ_1 is its middle point. The starred one, ζ_2 , is the global optimum instead.

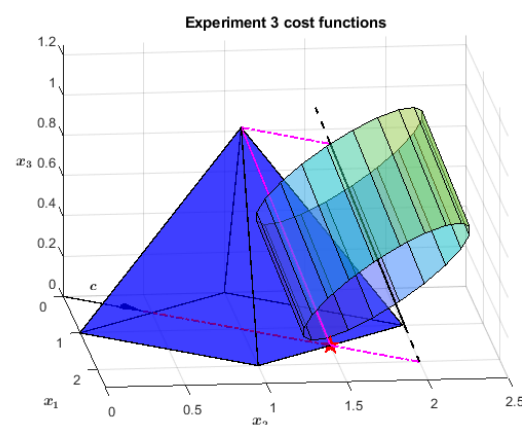


Figure 6. Cost functions: the primary is the distance from the oblique line, each infinite cylinder is a level surface; the secondary maximizes the sum of x_1 and x_2 .

Before reaching the global optimum, the NA-IPM passes close to the midpoint of the first objective optimal region. In the current problem, it is $\xi_1 = (\frac{5}{4}, \frac{5}{4}, \frac{1}{2})$, which is approached since iteration 3, as can be seen in Table 3. Then, starting from iteration 6, it goes towards ξ_2 since the algorithm perceives the first objective as fully optimized, as one can deduce from the change in magnitude of μ .

Table 3. Iterations of the NA-IPM solving the problem in (27).

iter	$\mu \in \mathbb{R}$	$x \in \mathbb{R}^3$	$f(x) \in \mathbb{E}$
0	3.37	[1.46 1.46 1.46]	$-31.72 - 2.92\eta$
1	1.23	[1.26 1.26 0.66]	$-30.64 - 2.52\eta$
2	0.11	[1.27 1.27 0.39]	$-29.71 - 2.54\eta$
3	2.12×10^{-3}	[1.30 1.30 0.39]	$-29.99 - 2.60\eta$
4	2.12×10^{-5}	[1.30 1.30 0.39]	$-30.00 - 2.61\eta$
5	2.12×10^{-7}	[1.30 1.30 0.39]	$-30.00 - 2.61\eta$
6	0.15η	[1.38 1.38 0.24]	$-30.00 - 2.76\eta$
7	0.06η	[1.50 1.50 0.00]	$-30.00 - 3.00\eta$
8	$1.14 \times 10^{-3}\eta$	[1.50 1.50 0.00]	$-30.00 - 3.00\eta$
9	$1.14 \times 10^{-5}\eta$	[1.50 1.50 0.00]	$-30.00 - 3.00\eta$
10	$1.14 \times 10^{-7}\eta$	[1.50 1.50 0.00]	$-30.00 - 3.00\eta$

5.4. Experiment 4: Three-Objective QP

The last test problem involves three distinct objective functions. Again, the step-by-step construction is reported in the appendix (Appendix B to be precise), while its standard and non-Archimedean descriptions are the ones in Equations (28) and (29), respectively:

$$\begin{aligned}
 &\text{lex min}_x \quad c^T x, \frac{1}{2}x^T Qx + q^T x, \frac{1}{2}x^T Px + p^T x \\
 &\text{s.t.} \quad \begin{aligned} &-x_1 + x_2 + x_3 \leq 1 \\ &-x_1 - x_2 + x_3 \leq 1 \\ &x_1 - x_2 + x_3 \leq 1 \\ &x_1 + x_2 + x_3 \leq 3 \\ &x_3 \geq 0 \end{aligned}
 \end{aligned} \tag{28}$$

$$\begin{aligned}
 &\min_x \quad \frac{1}{2}x^T(Q\eta + P\eta^2)x + (c + q\eta + p\eta^2)^T x \\
 &\text{s.t.} \quad \begin{aligned} &-x_1 + x_2 + x_3 \leq 1 \\ &-x_1 - x_2 + x_3 \leq 1 \\ &x_1 - x_2 + x_3 \leq 1 \\ &x_1 + x_2 + x_3 \leq 3 \\ &x_3 \geq 0 \end{aligned}
 \end{aligned} \tag{29}$$

where

$$c = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \quad Q = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad q = \begin{bmatrix} -5 \\ -5 \\ 0 \end{bmatrix}, \quad P = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad p = \begin{bmatrix} -5 \\ -3 \\ 2 \end{bmatrix}. \tag{30}$$

The feasible region is the same pyramid as in the previous problem, Section 5.3. This time, however, there are three objectives and only the first one is linear (see Figure 7). The latter promotes as optimal the whole farthest-from-origin face of the pyramid, the green one in Figure 8. The second objective is again a penalization for the distance from a line, which

in this case is $\mathcal{L} = (t, -t + \frac{5}{2}, 0)$, $t \in \mathbb{R}$. Of the original triangle, the only optimal surface for the second objective is the segment linking $(\frac{23}{12}, \frac{11}{12}, \frac{1}{6})$ and $(\frac{11}{12}, \frac{23}{12}, \frac{1}{6})$, highlighted in magenta in both the figures. The third objective selects just one point of that segment: $\zeta_3 = (\frac{5}{3}, \frac{7}{6}, \frac{1}{6})$. It consists of a potential function with the shape of an infinite paraboloid, centered on $(\frac{5}{4}, \frac{3}{4}, 0)$ and growing linearly along $\vec{x}_3$.

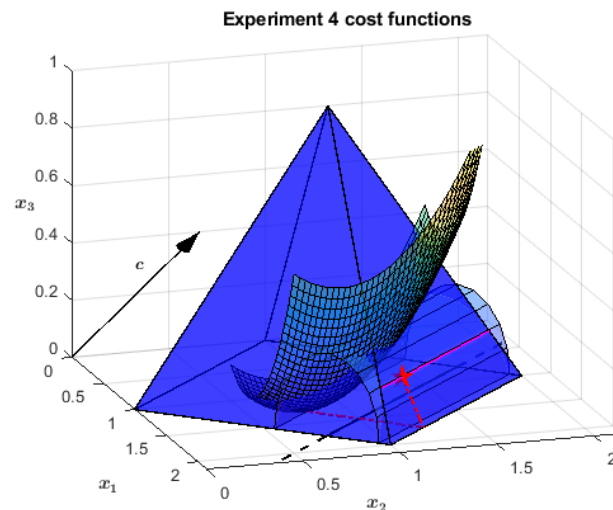


Figure 7. Cost functions: the primary objective is the vector c ; the secondary one is the distance from the black line on the x_1x_2 axis (the cylinder is a level surface); the tertiary one is the infinite paraboloid centered on $(\frac{5}{4}, \frac{3}{4}, 0)$.

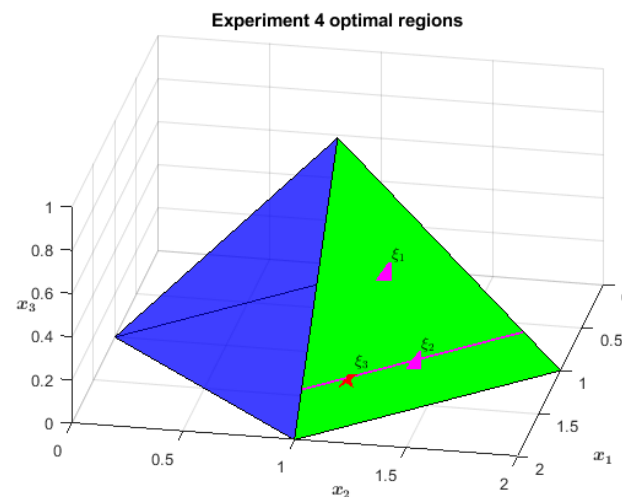


Figure 8. The surface in green is the optimal region of the first objective (ζ_1 is its midpoint); the segment in magenta is the also optimal region for the second objective (ζ_2 is its midpoint); ζ_3 is the global optimum.

As one can imagine, the NA-IPM passes close to two more points before reaching ζ_3 . The first of those is the center of the green triangle, i.e., $\zeta_1 = (\frac{5}{4}, \frac{5}{4}, \frac{1}{2})$. Then, it goes towards $\zeta_2 = (\frac{8}{3}, \frac{8}{3}, \frac{1}{6})$, which would have been one of the global optima (but the unique one with maximum centrality) if the problem had had just two objectives rather than three. Indeed, it is the midpoint of the second objective optimal segment along the green triangle. Table 4 identifies this switch in the optimizing direction at iterations 5–6, when μ becomes infinitesimal of the first order. Eventually, the third objective takes the stage and brings the optimization away from ζ_2 , pointing in the direction of ζ_3 . This happens at iterations 10–11, when the magnitude of μ decreases again, making it infinitesimal of the second order. Such

a value of μ testifies to the completion of both the first and second objective optimization, leaving space for the third.

Table 4. Iterations of the NA-IPM solving the problem in (28).

iter	$\mu \in \mathbb{R}$	$x \in \mathbb{R}^3$	$f(x) \in \mathbb{E}$
0	0.53	[1.46 1.46 1.46]	$-4.38 - 3.64\eta - 6.08\eta^2$
1	0.21	[1.32 1.32 0.74]	$-3.37 - 10.27\eta - 5.08\eta^2$
2	0.02	[1.30 1.30 0.40]	$-3.01 - 11.83\eta - 4.44\eta^2$
3	1.60×10^{-4}	[1.30 1.30 0.40]	$-3.00 - 11.84\eta - 4.44\eta^2$
4	1.60×10^{-6}	[1.30 1.30 0.40]	$-3.00 - 11.84\eta - 4.44\eta^2$
5	1.61×10^{-8}	[1.30 1.30 0.40]	$-3.00 - 11.84\eta - 4.44\eta^2$
6	0.06η	[1.38 1.38 0.25]	$-3.00 - 12.13\eta - 3.92\eta^2$
7	$2.21 \times 10^{-3}\eta$	[1.41 1.41 0.17]	$-3.00 - 12.17\eta - 3.66\eta^2$
8	$2.46 \times 10^{-5}\eta$	[1.42 1.42 0.17]	$-3.00 - 12.17\eta - 3.64\eta^2$
9	$2.48 \times 10^{-7}\eta$	[1.42 1.42 0.17]	$-3.00 - 12.17\eta - 3.64\eta^2$
10	$1.62 \times 10^{-9}\eta$	[1.42 1.42 0.17]	$-3.00 - 12.17\eta - 3.64\eta^2$
11	$0.14\eta^2$	[1.54 1.29 0.17]	$-3.00 - 12.17\eta - 3.82\eta^2$
12	$0.01\eta^2$	[1.65 1.19 0.17]	$-3.00 - 12.17\eta - 3.89\eta^2$
13	$1.63 \times 10^{-4}\eta^2$	[1.67 1.17 0.17]	$-3.00 - 12.17\eta - 3.89\eta^2$
14	$1.78 \times 10^{-6}\eta^2$	[1.67 1.17 0.17]	$-3.00 - 12.17\eta - 3.89\eta^2$
15	$1.59 \times 10^{-8}\eta^2$	[1.67 1.17 0.17]	$-3.00 - 12.17\eta - 3.89\eta^2$

5.5. An Informal Comparison With The Scalarization Approach

A typical standard approach to tackle (deterministic) lexicographic optimization problems is the scalarization method [18]. In terms of effectiveness, it is not able to guarantee equivalence between the original problem and the scalarized one, even if it converges to a solution significantly faster. More important, the choice of the scalarization parameters always requires a trial-and-error process to achieve good tuning: even in the case of a weighted combination of the objectives, the motto “the smaller the better” can be applied. Table 5 shows this fact quantitatively, using the benchmark in Section 5.4 as a reference problem.

Table 5. The approximation error of the optimal lexicographic point ξ_3 does not monotonically decrease when reducing the scalarization weight w on problem (28).

w	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
Error	6.59×10^{-3}	6.78×10^{-4}	8.17×10^{-5}	1.46×10^{-1}	3.29×10^{-1}

The study replaces the non-Archimedean weight η in (29) with a standard one (w), and solves the problem using the Matlab quadprog routine. The evidence highlights how values of w that are too large or too small introduce a bias in the optimization which negatively affects the computations. Actually, the error made by the solving algorithm, i.e., the norm of the discrepancy between the theoretically optimal point ξ_3 and the solution found, is not acceptable even when w decreases too much. Furthermore, the scalarization approach is always problem-dependent, which means it cannot constitute a general-purpose solving paradigm.

6. Conclusions

This paper builds upon a theoretical and practical achievement that happened in the last years concerning the solution of the lexicographic multi-objective linear programming problem. Indeed, in [4], it has been demonstrated that a non-Archimedean version of the Simplex algorithm is able to solve that problem, in an elegant and powerful way. The aim of this work was to verify whether or not a non-Archimedean version of an interior point algorithm (called the NA-IPM by the authors) is able to deliver a solution to the same class of problems and go beyond it. This paper answers affirmatively, and this is interesting for situations where an interior point method is known to perform better than Simplex, such as in high-dimensional problems. With this NA-IPM at hand, it was natural to verify if it was able to solve a convex lexicographic multi-objective quadratic programming problem, and again the answer is positive.

The proposed implementation of the NA-IPM shows a polynomial complexity, and its implementation guarantees to converge in finite time. The new algorithm also enjoys a very light embedding which gets rid of the issues related to infeasibility and unboundedness. As a practical application, this paper considered and discussed lexicographic optimization problems as well as infeasible/unbounded ones, testing the NA-IPM on four linear and quadratic programming examples, and achieving the expected results. It paves the way to more difficult ones, such as lexicographic multi-objective semi-definite programming problems, which are left as a future study. It should also be noted that the NA-IPM can be used to solve other families of problems involving infinitesimal/infinite numbers, as testified by Section 5.2. The study of its performances on harder examples is left for a future study as well.

Author Contributions: Conceptualization, L.F. and M.C.; Methodology, L.F.; Software, L.F.; Validation, L.F.; Formal analysis, L.F.; Investigation, L.F.; Data curation, L.F.; Writing—original draft, L.F.; Writing—review & editing, L.F. and M.C.; Visualization, L.F.; Supervision, M.C.; Project administration, M.C.; Funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

Acknowledgments: The authors wish to express our warmest appreciation to Vieri Benci for his generous help in teaching us Robinson's non-standard analysis and his Alpha Theory. The authors are very thankful to the three anonymous reviewers for their accurate comments and insightful remarks.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Cost Function Construction in Experiment 3

This appendix shows the construction of the cost function in problem (27). The second objective discussion is omitted since it is linear and, therefore, trivial.

The idea is to make the apothem of the farthest-from-origin face of the pyramid optimal for the first objective. This segment belongs to the line $\mathcal{G} = (t, t, -2t + 3)$, $t \in \mathbb{R}$. A possible cost function is the one that penalizes the distance from a line $\mathcal{L}$ parallel to $\mathcal{G}$. In addition, $\mathcal{L}$ must satisfy the property that no point in the primal feasible region (the pyramid) is closer to it than $\mathcal{G}$. A feasible choice is the translation of $\mathcal{G}$ along $\vec{x}_3$, say $\mathcal{L} = \mathcal{G} + [0, 0, 1] = (t, t, -2t + 4)$, $t \in \mathbb{R}$. In this case, the line versor is $v = (-\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{\sqrt{2}}{\sqrt{3}})$.

The distance d of any point $x \in \mathbb{R}^3$ from $\mathcal{L}$ comes from the following equation

$$d = x^1 - (v^T x^1)v, \quad (\text{A1})$$

where $x^1 = x - x^0$ and x^0 is any point on $\mathcal{L}$, i.e., $x^0 \in \mathcal{L}$. A few lines to explain the equation follow. The translation $x - x^0$ brings the system origin to x^0 , which allows one to consistently execute the projection of the point of interest (identified by x^1 in the new reference system) along $\mathcal{L}$. To do this, first, one computes the norm of the projection by the

inner product $v^T x^1$, then one constructs the projection by multiplying such norm by the line reference direction, that is its versor v . Since the projection of x^1 on $\mathcal{L}$ is parallel to it by definition, the difference between x^1 and such a projection must be orthogonal to it, i.e., it is the distance of x from $\mathcal{L}$ in the coordinate system with origin in x^0 .

In problem (27), x^0 is arbitrarily chosen as the vector $[2, 2, 0]^T$; therefore, the distance is:

$$d = \begin{bmatrix} x_1 - 2 \\ x_2 - 2 \\ x_3 \end{bmatrix} - \left(-\frac{1}{\sqrt{6}}(x_1 + x_2 - 4) + \frac{\sqrt{3}}{\sqrt{3}}x_3 \right) \begin{bmatrix} -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} \\ \frac{\sqrt{2}}{\sqrt{3}} \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 5x_1 + x_2 + 2x_3 - 8 \\ -x_1 + 5x_2 + 2x_3 - 8 \\ 2x_1 + 2x_2 + 2x_3 - 8 \end{bmatrix}.$$

To model the problem as a QP task, one can use the squared norm of the distance d as a cost function. After having done the calculations, its expression is (up to a multiplicative factor)

$$\|d\|^2 = 30x_1^2 + 30x_2^2 + 12x_3^2 - 12x_1x_2 + 24x_1x_3 + 24x_2x_3 - 96x_1 - 96x_2 - 96x_3 + k, \quad (\text{A2})$$

where k is a constant with no impact on the optimization and, therefore, it shall not be considered furthermore. Dividing (A2) by 6, one gets exactly the matrix Q and the vector q in (27) (we remind the reader that Q is divided by 2 in the cost function and, therefore, the diagonal entries must be doubled).

The optimal solution of the whole problem $\bar{x}$ must lie on the pyramid face apothem due to the first objective function, i.e., $\bar{x} = (\bar{t}, \bar{t}, -2\bar{t} + 3)$ for some $\bar{t} \in [1, \frac{3}{2}]$. The second objective, namely c in (27), forces $\bar{x}$ (read $\bar{t}$) to possess another property: it must also solve the following scalar optimization problem t :

$$\begin{aligned} \max_t \quad & 2t \\ \text{s.t.} \quad & t \in \left[1, \frac{3}{2}\right] \end{aligned}$$

whose optimal solution is $\bar{t} = \frac{3}{2}$. Therefore, the optimal solution of the whole problem is $\bar{x} = [\frac{3}{2}, \frac{3}{2}, 0]^T$, which coincides with ξ_2 in Section 5.3.

Appendix B. Cost Function Construction in Experiment 4

This appendix shows the construction of the cost function in problem (28). It consists of three different objectives, the first is linear while the other two are quadratic. Let us disclose them in order.

The first objective indicates as the optimal surface the whole farthest-from-origin face of the pyramidal feasible region. To select it, the cost vector c must be orthogonal to the plane π the triangular face belongs to. Actually, that plane is identified by the equation $x_1 + x_2 + x_3 = 3$, as testified by the fourth constraint in (28). Basic notions in linear algebra say that the normal direction to a plane is exactly the span of the vector filled with its coefficients, i.e., $[1, 1, 1]^T$. Since the problem is a minimization one and the constraint is lesser than or equal to one, then the normal vector must be reversed, that is $c = [-1, -1, -1]^T$, exactly as in (30).

Of such an optimal face, the second objective selects just a segment parallel to the basis. To do it, the distance from a straight line is used, similarly to what is done in Appendix A. Actually, the reference line must be parallel to both the basis and the face; $\mathcal{L} = (t, -t + \frac{5}{2}, 0)$, $t \in \mathbb{R}$ is a feasible choice. Indeed, it is parallel to the x_1x_2 plane (read the pyramid basis) since it is constant on x_3 , while it is parallel to the face because it is

parallel to its basis $\mathcal{B} = (t, -t + 3, 0)$, $t \in \mathbb{R}$. Using (A1), the distance equation in this case becomes:

$$d = \begin{bmatrix} x_1 - \frac{7}{4} \\ x_2 - \frac{3}{4} \\ x_3 \end{bmatrix} - \frac{1}{\sqrt{2}}(1 - x_1 + x_2) \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1 + x_2 - \frac{5}{2} \\ x_1 + x_2 - \frac{5}{2} \\ 2x_3 \end{bmatrix},$$

where $x^0 = [\frac{7}{4}, \frac{3}{4}, 0]^T$ and $v = [-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0]^T$. The squared norm of the distance is now (up to a multiplicative factor)

$$\|d\|^2 = x_1^2 + x_2^2 + 2x_3^2 + 2x_1x_2 - 5x_1 - 5x_2 + k \quad (\text{A3})$$

which originates Q and q of (30).

To analytically identify the optimal region for the second objective, let one project the problem on the $x_1 = x_2$ plane and then retrieve the whole surface, leveraging the fact that such surface is a segment parallel to $\mathcal{L}$. Applying the condition $x_1 = x_2 = t$, the pyramid face collapses to its apothem, whose equation is $\mathcal{G} = (t, t, -2t + 3)$, $t \in [1, \frac{3}{2}]$ (see Appendix A). Since the optimal region must belong to π , to retrieve it one has to substitute the equation of $\mathcal{G}$ into (A3), to differentiate with respect to t and to impose the optimality condition:

$$\begin{aligned} x \in \mathcal{G} \implies \|d\|^2 &= 12t^2 - 34t + k, \quad t \in \left[1, \frac{3}{2}\right], \\ \frac{d\|d\|^2}{dt} &= 24t - 34 = 0 \implies t = \frac{17}{12} \in \left[1, \frac{3}{2}\right]. \end{aligned}$$

Therefore, the optimal surface for the second objective is a line parallel to $\mathcal{L}$ and passing through $\xi = [\frac{17}{12}, \frac{17}{12}, \frac{1}{6}]^T$ (by construction it already belongs to π). Basic linear algebra considerations say that this request is equivalent to constraining the sought region to the line $\mathcal{S} = (t, -t + \frac{17}{6}, \frac{1}{6})$. To identify the values of t for which $\mathcal{S}$ belongs to the triangular face, one needs to intersect $\mathcal{S}$ with the pyramid edges $\mathcal{E}_0 = (t, 1, -t + 2)$ and $\mathcal{E}_1 = (1, t, -t + 2)$. The intersecting points are located at $t = \frac{11}{6}$ and $t = 1$, respectively. Therefore, the optimal region for the second objective is $\mathcal{S} = (t, -t + \frac{17}{6}, \frac{1}{6})$, $t \in [1, \frac{11}{6}]$, henceforth indicated by $\bar{\mathcal{S}}$.

The third objective is a potential function. By arbitrary choice, it is an infinite paraboloid centered in $[\frac{5}{4}, \frac{3}{4}, 0]^T$ and growing linearly along $\vec{x}_3$, i.e., its equation is (scaled up for practical reasons by a factor 2)

$$h(x) = 2 \left(\left(x_1 - \frac{5}{4} \right)^2 + \left(x_2 - \frac{3}{4} \right)^2 + x_3 \right) = 2x_1^2 + 2x_2^2 - 5x_1 - 3x_2 + 2x_3 + k \quad (\text{A4})$$

which coincides with P and p of (30).

The optimal point $\bar{x}$ of the whole problem is the point of $\bar{\mathcal{S}}$ (optimal surface identified by the second objective) which minimizes (A4). Since $\bar{x} \in \bar{\mathcal{S}}$, it has the form $\bar{x} = (\bar{t}, -\bar{t} + \frac{17}{6}, \frac{1}{6})$, $\bar{t} \in [1, \frac{11}{6}]$. Substituting this parametric description into (A4) and applying the first order optimality condition, one gets

$$x \in \bar{\mathcal{S}} \implies h(t) = 4t^2 - \frac{40}{3}t + k, \quad t \in [1, \frac{11}{6}] \quad \frac{dh}{dt} = 8t - \frac{40}{3} = 0 \implies \bar{t} = \frac{5}{3} \in [1, \frac{11}{6}].$$

Therefore, $\bar{x} = [\frac{5}{3}, \frac{7}{6}, \frac{1}{6}]^T$.

References

1. Bazaraa, M.S.; Sherali, H.D.; Shetty, C.M. *Nonlinear Programming: Theory and Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013. [CrossRef]

2. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004 [[CrossRef](#)]
3. Gondzio, J. Interior point methods 25 years later. *Eur. J. Oper. Res.* **2012**, *218*, 587–601. [[CrossRef](#)]
4. Cococcioni, M.; Pappalardo, M.; Sergeyev, Y.D. Lexicographic multi-objective linear programming using grossone methodology: Theory and algorithm. *Appl. Math. Comput.* **2018**, *318*, 298–311. [[CrossRef](#)]
5. Khorram, E.; Zarepisheh, M.; Ghaznavi-ghosoni, B. Sensitivity analysis on the priority of the objective functions in lexicographic multiple objective linear programs. *Eur. J. Oper. Res.* **2010**, *207*, 1162–1168. [[CrossRef](#)]
6. Letsios, D.; Mistry, M.; Misener, R. Exact lexicographic scheduling and approximate rescheduling. *Eur. J. Oper. Res.* **2021**, *290*, 469–478. [[CrossRef](#)]
7. Pourkarimi, L.; Zarepisheh, M. A dual-based algorithm for solving lexicographic multiple objective programs. *Eur. J. Oper. Res.* **2007**, *176*, 1348–1356. [[CrossRef](#)]
8. Sherali, H.D. Equivalent weights for lexicographic multi-objective programs: Characterizations and computations. *Eur. J. Oper. Res.* **1982**, *11*, 367–379. [[CrossRef](#)]
9. Hwang, C.L.; Masud, A.S.M. *Multiple Objective Decision Making—Methods and Applications: A State-of-the-Art Survey*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 164. [[CrossRef](#)]
10. Arora, J. Multiobjective Optimum Design Concepts and Methods. In *Introduction to Optimum Design*; Elsevier: Amsterdam, The Netherlands, 2012; pp. 657–679. [[CrossRef](#)]
11. Benci, V.; Cococcioni, M. The Algorithmic Numbers in Non-Archimedean Numerical Computing Environments. *Discret. Contin. Dyn. Syst.-Ser.* **2021**, *14*, 1673. [[CrossRef](#)]
12. Benci, V.; Cococcioni, M.; Fiaschi, L. Non-Standard Analysis Revisited: An Easy Axiomatic Presentation Oriented Towards Numerical Applications. *Int. J. Appl. Math. Comput. Sci.* **2021**, *32*, 65–80. [[CrossRef](#)]
13. Sergeyev, Y.D. Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems. *EMS Surv. Math. Sci.* **2017**, *4*, 219–320. [[CrossRef](#)]
14. Cococcioni, M.; Fiaschi, L.; Lambertini, L. Non-Archimedean Zero Sum Games. *Appl. Comput. Appl. Math.* **2021**, *393*, 113483. [[CrossRef](#)]
15. Fiaschi, L.; Cococcioni, M. Non-Archimedean Game Theory: A Numerical Approach. *Appl. Math. Comput.* **2020**, *409*, 100687. [[CrossRef](#)]
16. Lai, L.; Fiaschi, L.; Cococcioni, M. Solving Mixed Pareto-Lexicographic Many-Objective Optimization Problems: The Case of Priority Chains. *Swarm Evol. Comput.* **2020**, *55*. [[CrossRef](#)]
17. Lai, L.; Fiaschi, L.; Cococcioni, M.; Deb, K. Solving Mixed Pareto-Lexicographic Many-Objective Optimization Problems: The Case of Priority Levels. *IEEE Trans. Evol. Comput.* **2021**, *25*, 971–985. [[CrossRef](#)]
18. Lai, L.; Fiaschi, L.; Cococcioni, M.; Kalyanmoy, D. Pure and mixed lexicographic-Paretian many-objective optimization: state of the art. *Nat. Comput.* **2022**, *1*, 1–16. [[CrossRef](#)]
19. 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. *Comm. Nonlinear Sci. Numer. Simul.* **2020**, *84*, 105177. [[CrossRef](#)]
20. Cococcioni, M.; Fiaschi, L. The Big-M Method using the Infinite Numerical M. *Optim. Lett.* **2021**, *15*, 2455–2468. [[CrossRef](#)]
21. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006. [[CrossRef](#)]
22. Kozlov, M.K.; Tarasov, S.P.; Khachiyan, L.G. Polynomial solvability of convex quadratic programming. *Dokl. Akad. Nauk.* **1979**, *248*, 1049–1051. [[CrossRef](#)]
23. Forsgren, A.; Gill, P.E.; Wright, M.H. Interior methods for nonlinear optimization. *SIAM Rev.* **2002**, *44*, 525–597. [[CrossRef](#)]
24. Mehrotra, S. On the implementation of a primal-dual interior point method. *SIAM J. Optim.* **1992**, *2*, 575–601. [[CrossRef](#)]
25. Gondzio, J.; Grothey, A. Direct solution of linear systems of size 10^9 arising in optimization with interior point methods. In *International Conference on Parallel Processing and Applied Mathematics*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 513–525. [[CrossRef](#)]
26. Colombo, M.; Gondzio, J. Further development of multiple centrality correctors for interior point methods. *Comput. Optim. Appl.* **2008**, *41*, 277–305. [[CrossRef](#)]
27. Wright, S.J. *Primal-Dual Interior-Point Methods*; SIAM: Philadelphia, PA, USA, 1997. [[CrossRef](#)]
28. Andersen, E.D.; Andersen, K.D. Presolving in linear programming. *Math. Program.* **1995**, *71*, 221–245. [[CrossRef](#)]
29. Bertsekas, D.P. Nonlinear programming. *J. Oper. Res. Soc.* **1997**, *48*, 334–334. [[CrossRef](#)]
30. Stoer, J.; Bulirsch, R. *Introduction to Numerical Analysis*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 12. [[CrossRef](#)]
31. Gonzaga, C.C. Path-following methods for linear programming. *SIAM Rev.* **1992**, *34*, 167–224. [[CrossRef](#)]
32. Kojima, M.; Megiddo, N.; Mizuno, S. A general framework of continuation methods for complementarity problems. *Math. Oper. Res.* **1993**, *18*, 945–963. [[CrossRef](#)]
33. Zhang, S. A new self-dual embedding method for convex programming. *J. Glob. Optim.* **2004**, *29*, 479–496. [[CrossRef](#)]
34. Benci, V.; Di Nasso, M. *How to Measure the Infinite: Mathematics with Infinite and Infinitesimal Numbers*; World Scientific: Singapore, 2018. [[CrossRef](#)]
35. Keisler, H.J. *Foundations of Infinitesimal Calculus*; Prindle, Weber & Schmidt Boston: Boston, MA, USA, 1976; Volume 20.

36. Rossi, F.; Fiaschi, L.; Cococcioni, M.; Saponara, S. Design and FPGA Synthesis of BAN Processing Unit for non-Archimedean Number Crunching. In *Applications in Electronics Pervading Industry, Environment and Society*; Saponara, S., De Gloria, A., Eds.; Springer International Publishing: Cham, Switzerland, 2022.
37. Megiddo, N. Pathways to the optimal set in linear programming. In *Progress in Mathematical Programming*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 131–158. [[CrossRef](#)]
38. Kojima, M.; Mizuno, S.; Yoshise, A. A primal-dual interior point algorithm for linear programming. In *Progress in Mathematical Programming*; Springer: Berlin/Heidelberg, Germany, 1989; pp. 29–47. [[CrossRef](#)]
39. Monteiro, R.D.; Adler, I. Interior path following primal-dual algorithms. Part I: Linear programming. *Math. Program.* **1989**, *44*, 27–41. [[CrossRef](#)]
40. Monteiro, R.D.; Adler, I. Interior path following primal-dual algorithms. Part II: Convex quadratic programming. *Math. Program.* **1989**, *44*, 43–66. [[CrossRef](#)]
41. McShane, K.A.; Monma, C.L.; Shanno, D. An implementation of a primal-dual interior point method for linear programming. *ORSA J. Comput.* **1989**, *1*, 70–83. [[CrossRef](#)]
42. Lustig, I.J. Feasibility issues in a primal-dual interior-point method for linear programming. *Math. Program.* **1990**, *49*, 145–162. [[CrossRef](#)]
43. Lustig, I.J.; Marsten, R.E.; Shanno, D.F. Computational experience with a primal-dual interior point method for linear programming. *Linear Algebra Its Appl.* **1991**, *152*, 191–222. [[CrossRef](#)]