Fundamental Limits for Iterated Function Optimization on Turing Machines

Holger Boche[†], Volker Pohl[†] and H. Vincent Poor[‡]
[†] Chair of Theoretical Information Technology, Technical University of Munich, Germany, {volker.pohl, boche}@tum.de
[‡] Department of Electrical and Computer Engineering, Princeton University, Princeton, USA, poor@princeton.edu

***Abstract*—This paper studies the effective convergence of iterative methods for solving convex minimization problems using block Gauss–Seidel algorithms. It investigates whether it is always possible to algorithmically terminate the iteration in such a way that the outcome of the iterative algorithm satisfies any predefined error bound. It is shown that the answer is generally negative. Specifically, it is shown that even if a computable continuous function which is convex in each variable possesses computable minimizers, a block Gauss–Seidel iterative method might not be able to effectively compute any of these minimizers. This means that it is impossible to algorithmically terminate the iteration such that a given performance guarantee is satisfied. The paper discusses two reasons for this behavior and gives simple and concrete examples.**

I. Introduction

Many problems in engineering and science, e.g. in communications, signal processing, information theory, artificial intelligence, control, finance, can be formulated as optimization problems (see, e.g. [1]–[6]). In applications, however, the concrete structure of the problem makes it often impossible to perform a global optimization of all performance functions and over all parameters. Therefore, iterative methods are applied.

Modern communication systems provide an important example of this situation. These systems use a bidirectional communication protocol and have a highly distributed and hierarchical structure with many different communication notes such as base stations, access points, and individual users. The performance functions (e.g. the transmission rates) of the individual communication notes depend on the signal processing and coding parameters (e.g. beam forming coefficients, MIMO antenna gains, transmit powers, frequency dependent channel parameters, waveforms, etc.) of all other communication notes. This decentralized structure together with the multiuser interference, introduced by the wireless communication channel, imply that a joint optimization of all performance functions based on all parameters of the system is practically impossible. Therefore individual, decentralized optimization strategies, that iteratively optimize only over a subset of the parameters, have been proposed and applied for such systems. In fact, already the first multiuser beamformaing approaches [7], [8] use iterative optimization. In distributed multiuser MIMO systems, the number of parameters grows further [9]–[12], and modern applications like integrated sensing and communications require even more complex performance functions because of the joint optimization of communication and pulse forming for the radar [13], [14]. Consequently, also these systems rely on iterative optimization methods.

Modern computer technology allows to tackle huge optimization problems in very high dimensions. Nonetheless, up to now, there exists no general rule to programmatically interrupt an iterative optimization procedure such that the result of the computation is guaranteed to satisfy a predefined error bound with respect to the true solution. As an example, we mention the fairly simple problem of approximating the optimal input distribution that achieves the capacity of discrete memoryless channels [15]–[17]. Using techniques from computability theory, this paper will actually show that such a stopping rule can not exist for iterative optimization problems, in general.

Immediately after the precise formulation of the mathematical notion of *computing*, due to Turing [18], [19], it became clear that few physical problems can be solved exactly by digital computations. Most solutions can only be approximated (arbitrarily well). Therefore Turing's notion of computation requires that the result of a computation has to satisfy a predefined error bound. It turns out that there exist computational objects (numbers, functions, etc.) for which this is not possible. Then they are said to be *non-computable*. If such an error control is possible, the objects are *computable*. In this paper, we investigate the ability to solve optimization problems on digital hardware. To this end, we consider optimization problems in the m-dimensional real Euclidean space $\mathbb{R}^m$:

$$\min_{\mathbf{x}\in\mathcal{R}} f(\mathbf{x}) \tag{1}$$

where $f : \mathbb{R}^m \to \mathbb{R}$ is a continuous function and $\mathcal{R} \subset \mathbb{R}^m$ is a convex, compact subset of $\mathbb{R}^m$. It is well known [20] that the minimum value $\mathrm{Min}_{\mathcal{R}}(f) = \min_{\mathbf{x}\in\mathcal{R}} f(\mathbf{x})$ of (1) is always computable. If (1) has a unique minimizer, i.e. if the set

$$\mathcal{MIN}_{\mathcal{R}}(f) = \big\{\widehat{\mathbf{x}} \in \mathcal{R} : f(\widehat{\mathbf{x}}) = \mathrm{Min}_{\mathcal{R}}(f)\big\} \tag{2}$$

contains exactly one vector, then also this minimizer is computable. If, on the other hand, the minimizer is not unique then some or all minimizers might not be computable [20]. So as long as (1) has at least one computable minimizer, this minimizer can, in principle, algorithmically be computed on a digital computer. One only needs to construct a sequence $\big\{\mathbf{x}^{(k)}\big\}_{k\in\mathbb{N}} \subset \mathbb{R}^m$ that effectively converges to $\widehat{\mathbf{x}}$. Since $\widehat{\mathbf{x}}$ is computable, this is always possible. However, finding such a sequence might be a fairly complicated and creative problem and the question is whether this process can (efficiently) be automated on a digital computer.

Preprint: Final version in Proc. *2025 IEEE International Symposium on Information Theory (ISIT)*, Ann Arbor, MI, USA, June 22–27, 2025.

Practical algorithms often apply a certain (suboptimal) strategy to find a sequence $\left\{\mathbf{x}^{(k)}\right\}_{k\in\mathbb{N}}$ that converges to a minimizer of f. One strategy is to minimize successively only over some dimensions of $\mathbf{x}\in\mathbb{R}^m$ while keeping the other dimensions fixed: For example, let

$$\mathcal{R} = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_m, b_m] \tag{3}$$

and let $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_m^{(0)}) \in \mathbb{R}^m$ be an arbitrary initialization vector. Then one successively solves for $\ell = 1, 2, \ldots, m$ the one dimensional optimization problems

$$x_\ell^{(k+1)} = \arg\min_{y\in[a_\ell, b_\ell]} f(x_1^{(k+1)}, \ldots, x_{\ell-1}^{(k+1)}, y, x_{\ell+1}^{(k)}, \ldots, x_m^{(k)})$$

and iterates over $k = 0, 1, 2, \ldots$. This strategy produces a sequence $\left\{\mathbf{x}^{(k)}\right\}_{k\in\mathbb{N}}$ that converges to a global minimizer $\widehat{\mathbf{x}}$ of f under fairly weak (convexity) conditions on f. However, the so determined sequence $\left\{\mathbf{x}^{(k)}\right\}_{k\in\mathbb{N}}$ may not *effectively* converge to the global minimizer. Instead, it may happen that:

- Some of the computational steps are not computable, i.e. they may not be realizable on a digital computer with an effective control of the approximation error.
- The sequence $\left\{\mathbf{x}^{(k)}\right\}_{k\in\mathbb{N}}$ may converge to a non-computable minimizer of f.

Thus even through there exists a global computable optimizer of f, the automated procedure might not be able to find a corresponding approximation sequence $\left\{\mathbf{x}^{(k)}\right\}_{k\in\mathbb{N}}$ of computable vectors that effectively converges to this minimizer, i.e. the automated procedure might not be able to effectively compute a global minimizer of f.

This paper studies the described iterative optimization strategy. We will show that there exist simple (piecewise linear) functions f that are convex in each coordinate, that have infinitely many computable minimizers but such that the iterative optimization is not able to find a computable sequence that converges effectively to any of the computable minimizers.

Because of space constraints, this paper contains no formal proofs. These proofs, together with more results and discussions can be found in the preprint [21].

II. Notation and Computability

Vectors in $\mathbb{R}^m$ are denoted by boldface lower-case letters written as row vectors like $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. As usual, $[a, b] = \{x \in \mathbb{R} : a \le x \le b\}$ denotes a closed interval on $\mathbb{R}$. For real numbers $a_i < b_i$, $i = 1, 2, \ldots, m$, (3) is said to be a *rectangle* in $\mathbb{R}^m$. The set of all continuous functions on $\mathbb{R}^m$ or on $\mathcal{R} \subset \mathbb{R}^m$ are denoted by $\mathcal{C}(\mathbb{R}^m)$ or $\mathcal{C}(\mathcal{R})$, respectively.

Computability analysis: We briefly review concepts and notions of computability analysis [18]–[20], [22]–[24]. It central concept is the notion of *effective convergence*:

Definition 1: *Let $\boldsymbol{x} = \{x_n\}_{n\in\mathbb{N}}$ be a sequence of real numbers that converges to $x \in \mathbb{R}$. We say that $\boldsymbol{x}$ converge effectively to x if $|x - x_n| \le 2^{-n}$ for all $n \in \mathbb{N}$.*

For an effectively convergent sequence, one can control the approximation error, i.e. for arbitrary small $\epsilon = 2^{-n}$ ´ one can determine $n \in \mathbb{N}$ so that $|x - x_n|$ is less than ϵ.

Every $x \in \mathbb{R}$ is the limit of a sequence of rational numbers. Only if this sequence effectively converges, x is said to be computable.

Definition 2: *A number $x \in \mathbb{R}$ is said to be* computable *if there exists a sequence $\{r_k\}_{k\in\mathbb{N}} \subset \mathbb{Q}$ that converges effectively to x. A vector $\mathbf{x} \in \mathbb{R}^m$ is said to be computable if each of its components is a computable number.*

We write $\mathbb{R}_{\mathrm{c}} \subsetneq \mathbb{R}$ for the subfield of all computable real numbers and $\mathbb{R}_{\mathrm{c}}^m$ for the set of computable vectors in $\mathbb{R}^m$. A rectangle $\mathcal{R} \subset \mathbb{R}^m$ as in (3) is said to be *computable* if all a_i, b_i, $i = 1, 2, \ldots, m$ are computable numbers.

Definition 3: *A sequence $\boldsymbol{x} = \{x_n\}_{n\in\mathbb{N}}$ of real numbers is said to be* computable *if there exists a doubly indexed rational sequence $\{r_{n,k}\}_{n,k\in\mathbb{N}} \subset \mathbb{Q}$ such that for every $n \in \mathbb{N}$*

$$|x_n - r_{n,k}| < 2^{-k}, \quad \textit{for all } k \in \mathbb{N}.$$

A sequence $\{\mathbf{x}_n\}_{n\in\mathbb{N}} \subset \mathbb{R}^m$ is said to be computable if each component of this sequence is a computable sequence.

Besides computable numbers and sequences, *computable functions* will play an important role in this paper.

Definition 4: *Let $m, M \in \mathbb{N}$. A function $f : \mathbb{R}^m \to \mathbb{R}^M$ is said to be* Banach–Mazur computable*, if for every computable sequences $\{\mathbf{x}_n\}_{n\in\mathbb{N}} \subset \mathbb{R}^m$ the sequence $\{f(\mathbf{x}_n)\}_{n\in\mathbb{N}} \subset \mathbb{R}^M$ is computable.*

Definition 5: *A function $f : \mathcal{R} \to \mathbb{R}$, on a computable rectangle $\mathcal{R} \subset \mathbb{R}^m$, is said to be* effectively uniformly continuous *if there exits a recursive function $\mathrm{d} : \mathbb{N} \to \mathbb{N}$ such that for every $k \in \mathbb{N}$ and $n = \mathrm{d}(k)$*

$$\|\mathbf{x}_1 - \mathbf{x}_2\| \le 2^{-n} \quad \textit{implies} \quad |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \le 2^{-k}.$$

A function that is Banach–Mazur computable and effectively uniformly continuous is said to be a computable continuous function*, and $\mathcal{C}_{\mathrm{c}}(\mathcal{R})$ denotes the set of computable continuous functions on $\mathcal{R}$.*

Remark: There exist other notions of computable functions (see, e.g., [25] for an overview and its logical relations). Here we only need that every function that is not Banach–Mazur computable is also not a computable continuous function.

III. Optimization of Smooth Functions

This section explains in more detail the iterative optimization algorithm that will be studied in this paper and introduces some more notation needed to formulate our main results.

a) Minimum value and minimizer: We consider general minimization problems with a so-called box constraint [26]: Let $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2 \times \cdots \times \mathcal{R}_r \subset \mathbb{R}^m$ be a computable rectangle with $\mathcal{R}_\ell \subset \mathbb{R}^{n_\ell}$ and $\sum_{\ell=1}^r n_\ell = m$, and let $f : \mathcal{R} \to \mathbb{R}$ be a continuous function on $\mathcal{R}$. Now we may ask for the *minimum value* of f on the rectangle $\mathcal{R}$, i.e. for the value

$$\mathrm{Min}_{\mathcal{R}}(f) = \min_{\mathbf{x}\in\mathcal{R}} f(\mathbf{x}) = \min_{\mathbf{x}_\ell\in\mathcal{R}_l, 1\le\ell\le r} f(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_r), \tag{4}$$

and we may ask for a corresponding *minimizer*, i.e. for a vector

$$\widehat{\mathbf{x}} \in \mathcal{R} \quad \text{such that} \quad f(\widehat{\mathbf{x}}) = \min_{\mathbf{x}\in\mathcal{R}} f(\mathbf{x}). \tag{5}$$

Note that the minimizer is generally not unique but there may exist a whole set (2) of *global minimizers* in $\mathcal{R}$. It depends on the actual application whether one needs to find the minimum value or the minimizer, but it is clear that if one knows a minimizer $\widehat{\mathbf{x}}$ then one also knows the minimum value $\mathrm{Min}_{\mathcal{R}}(f) = f(\widehat{\mathbf{x}})$. Conversely, knowing $\mathrm{Min}_{\mathcal{R}}(f)$ may not help in finding the corresponding minimizer $\widehat{\mathbf{x}}$.

Remark: Instead of the minimization problem (4), one may consider a corresponding maximization problem, i.e. the problem finding the maximum value of f on $\mathcal{R}$ or of finding the maximizer of f. Such a maximization problem can always be transformed into a minimization problem by considering the function $-f$ on $\mathcal{R}$. Then the minimizer of $-f$ is the maximizer of f. Therefore, without loss of generality, this paper only discusses the minimization problem (4).

b) Computability of the minimum value and the minimizer: Apart from very special cases, there exists no closed-form solution for $\mathrm{Min}_{\mathcal{R}}(f)$ or $\widehat{\mathbf{x}}$. Therefore these values are usually approximated using numerical algorithms that determine a sequence that is guaranteed to converge to the optimal value. For a wide variety of optimization problems, algorithms are known that converge to the minimum value or minimizer, respectively. Most notable are certainly algorithms for convex optimization [26]–[28]. Nevertheless, from a practical point of view, the question is not only whether the algorithm converges to the optimum but whether this convergence is *effective*, i.e. whether the minimum value or the minimizer is computable. With respect to the computation of the minimum value $\mathrm{Min}_{\mathcal{R}}(f)$ the following result concerning its computability is well known (cf., e.g., [20, Chapter 6]).

Proposition 1: *Let $\mathcal{R}_\ell \subset \mathbb{R}^{n_\ell}$, $\ell = 1, 2, \ldots, r$, be computable rectangles, and let $\mathcal{R} = \mathcal{R}_1 \times \cdots \times \mathcal{R}_r \subset \mathbb{R}^m$. There exists a Turing machine $\mathrm{TM}_{\mathrm{Min}}$ that computes for every computable continuous function $f \in \mathcal{C}_{\mathrm{c}}(\mathcal{R})$ the value $\mathrm{Min}_{\mathcal{R}}(f)$.*

So the minimum value (4) is always algorithmically computable on a digital computer provided f is a computable continuous function. Moreover, $\mathrm{TM}_{\mathrm{Min}}$ in Proposition 1 is *universal* in the sense that it only depends on the rectangle $\mathcal{R}$. So for a fixed $\mathcal{R}$, the corresponding $\mathrm{TM}_{\mathrm{Min}}$ can compute $\mathrm{Min}_{\mathcal{R}}(f)$ for all $f \in \mathcal{C}_{\mathrm{c}}(\mathcal{R})$ as input [6].

For the computation of the minimizer $\widehat{\mathbf{x}}$, it is known that if (4) has a unique (global) minimizer then this minimizer is computable (cf., e.g., [20, Chap. $I.0.6$]). If the minimizer is not unique then some (or all) minimizers might not be computable and there exist several examples of computable continuous functions f that attain there minimum only at non-computable points (see, e.g., [29] and references in [20]).

c) Iterative optimization: Minimizing jointly over all m components of $\mathbf{x} \in \mathcal{R} \subset \mathbb{R}^m$ in (4), is often considered as being too complex. Therefore block coordinate optimization methods of the Gauss–Seidel type are applied that iteratively optimize over sub-rectangles $\mathcal{R}_\ell$, $\ell = 1, 2, \ldots, r$ while keeping the other variables fixed [30]–[34]. Starting with an initial guess $\widetilde{\mathbf{x}}^{(0)} = (\widetilde{\mathbf{x}}_1^{(0)}, \widetilde{\mathbf{x}}_2^{(0)}, \ldots, \widetilde{\mathbf{x}}_r^{(0)}) \in \mathcal{R}$, one solves for $k = 0$ the optimization problems

$$\widetilde{\mathbf{x}}_\ell^{(k+1)} = \arg\min_{\mathbf{y} \in \mathcal{R}_\ell} f\big(\widetilde{\mathbf{x}}_1^{(k+1)}, \ldots, \widetilde{\mathbf{x}}_{\ell-1}^{(k+1)}, \mathbf{y}, \widetilde{\mathbf{x}}_{\ell+1}^{(k)}, \ldots, \widetilde{\mathbf{x}}_r^{(k)}\big) \tag{6}$$

successively for $\ell = 1, 2, \ldots, r$ and iterates over $k = 1, 2, \ldots$. In particular, if $n_1 = n_2 = \ldots = n_\ell = 1$ then each step optimizes only over one coordinate of the vector $\widetilde{\mathbf{x}}^{(k)} \in \mathbb{R}^m$. This procedure yields a sequence $\big\{\widetilde{\mathbf{x}}^{(k)}\big\}_{k \in \mathbb{N}}$ of approximations of a minimizer of (4). The components of each vector $\widetilde{\mathbf{x}}^{(k)}$ are minimizers of a local optimization problem (6). Therefore $\big\{\widetilde{\mathbf{x}}^{(k)}\big\}_{k \in \mathbb{N}}$ is said to be a *sequence of local minimizers*.

One can show that under mild conditions on f, this sequence of local minimizers converges to a global minimizer $\widehat{\mathbf{x}}$ of (4). In fact, there exist many studies investigating the convergence of iterative algorithms for optimization problems of the form (1) (see, e.g., [33], [34]). These works investigate the convergence of sequences $\big\{\widetilde{\mathbf{x}}^{(k)}\big\}_{k \in \mathbb{N}}$ obtained by an iterative optimization algorithm. Apart from results regarding the convergence, there seems to exist no estimates on the convergence speed of these algorithms. However, such results are highly desirable from a practical point of view because the convergence results alone imply, in principle, that the iterative algorithm has to compute *ad infinitum* to reach the optimal value. In practice, one needs a criterion to stop the iteration if a desired error bound is achieved, i.e. one needs the possibility to pass an $M \in \mathbb{N}$ to the algorithm. Then the algorithm should be able to stop the iteration at $K \in \mathbb{N}$ if $\big|\widehat{\mathbf{x}} - \widetilde{\mathbf{x}}^{(K)}\big| < 2^{-M}$. Up to now, no such algorithm is known which brings us to our first question:

Question 1: *Given a sequence $\big\{\widetilde{\mathbf{x}}^{(k)}\big\}_{k \in \mathbb{N}}$ of local minimizers that converges to a global minimizer $\widehat{\mathbf{x}}$ of* (4)*. Is this convergence always effective?*

We will show that the answer is generally "no", even in the case $m = 2$. This implies that there can be no algorithmic stopping criterion for such iterative optimization algorithms.

Remark: The non-existence of such an algorithmic stopping criterion was recently observed for several central problems in information theory. One example is the celebrated *Blahut–Arimoto algorithm* [15], [16]. Further information theoretic questions showing a similar behavior are discussed in [36].

The $\arg\min$-operation in step (6) of the iterative algorithm is only a pseudo-code and it is not clear at the outset whether there exists an effective implementation on a digital computer for this operation. This raises the following general question.

Question 2: *Does there always exists an effective implementation of Step* (6)*?*

Associated with the block coordinate optimization method (6), we define for every $\ell = 1, 2, \ldots, r$, the sets

$$\begin{aligned}\mathcal{MIN}_\ell &= \mathcal{MIN}_\ell(\mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_r) \\ &= \Big\{\widehat{\mathbf{x}}_\ell \in \mathcal{R}_\ell \ : f(\mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}, \widehat{\mathbf{x}}_\ell, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_r) = \\ &\qquad \min_{\mathbf{y} \in \mathcal{R}_\ell} f(\mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}, \mathbf{y}, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_r)\Big\} \end{aligned} \tag{7}$$

of all *local minimizes* with respect to the ℓth coordinate, i.e. $\mathcal{MIN}_\ell$ contains all $\widehat{\mathbf{x}} \in \mathcal{R}_\ell$ that minimize the right hand side of (6). Then the $\arg\min$-operator in (6) simply chooses one element from $\mathcal{MIN}_\ell$. This operation can be described by a so-called assignment function:

Definition 6: *Consider the optimization problem* (4) *for a function* $f : \mathcal{R} \to \mathbb{R}$. *A function* $G_\ell : \mathcal{R}_1 \times \cdots \times \mathcal{R}_{\ell-1} \times \mathcal{R}_{\ell+1} \times \cdots \times \mathcal{R}_r \to \mathcal{R}_\ell$ *is said to be an* assignment function of f *for the* ℓ*th coordinate of the iterative optimization procedure if it has the property*

$$G_\ell(\mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}, \mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_r) \in \mathcal{MIN}_\ell\,. \tag{8}$$

The set of all assignment functions of f *for the* ℓ*th coordinate will be denoted by* $\mathcal{A}_\ell(f)$.

With this notion, step (6) can be written as

$$\mathbf{x}_\ell^{(k+1)} = G_\ell\big(\mathbf{x}_1^{(k+1)}, \ldots, \mathbf{x}_{\ell-1}^{(k+1)}, \mathbf{x}_{\ell+1}^{(k)}, \ldots, \mathbf{x}_r^{(k)}\big)\,. \tag{9}$$

In particular, the coordinate-wise optimization where each $\mathcal{R}_\ell$ is one-dimensional, can be rewritten as in Algorithm 1.

Algorithm 1: Coordinate-wise optimization

Initialization: $\mathbf{x}^0 = (x_1^{(0)}, \cdots, x_m^{(0)}) \in \mathbb{R}^m$, $k = 0$
repeat
  forall $\ell = 1, 2, \ldots, m$ **do**
    $x_\ell^{(k+1)} = G_\ell(x_1^{(k+1)}, \ldots, x_{\ell-1}^{(k+1)}, x_{\ell+1}^{(k)}, \ldots)$
  $k = k + 1$
until *convergence*;
Output : $\mathbf{x}_{\text{out}} = \mathbf{x}^{(k)}$

For every $\ell \in \{1, 2, \ldots, r\}$ the set $\mathcal{A}_\ell(f)$ contains generally many different assignment functions G_ℓ. Namely, there are as many different G_ℓ as there are different vectors in $\mathcal{MIN}_\ell$. In principle, one can choose any $G_\ell \in \mathcal{A}_\ell(f)$ for the optimization step (9). However, in order that step (6) be algorithmically solvable, we need to choose $G_\ell \in \mathcal{A}_\ell(f)$ to be a computable function. The interesting question is whether this is always possible. The next section will show that there are very simple computable continuous functions f such that for some $\ell \in \{1, 2, \ldots, r\}$, the set $\mathcal{A}_\ell(f)$ contains no computable assignment function.

IV. Computability of Assignment Functions

To make our arguments and examples as clear as possible, we consider the simplest case of the optimization problem (4), namely we consider functions $f : \mathbb{R}^2 \to \mathbb{R}$ on the rectangle $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ with $\mathcal{R}_1 = [-a, a]$ and $\mathcal{R}_2 = [-b, b]$ for arbitrary positive computable numbers $a, b \in \mathbb{R}_{\rm c}$. Then the corresponding iterative optimization algorithm is a coordinate-wise optimization as shown in Algorithm 1.

We show first that there exist computable continuous functions f_1 of two variables such that the set $\mathcal{A}_1(f_1)$ of assignment functions contains no computable assignment function.

Theorem 2: *Let* $a, b \in \mathbb{R}_{\rm c}$ *be arbitrary positive numbers and let* $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ *with* $\mathcal{R}_1 = [-a, a]$ *and* $\mathcal{R}_2 = [-b, b]$. *There exist computable continuous functions* $f_1 : \mathcal{R} \to \mathbb{R}$ *such that:*

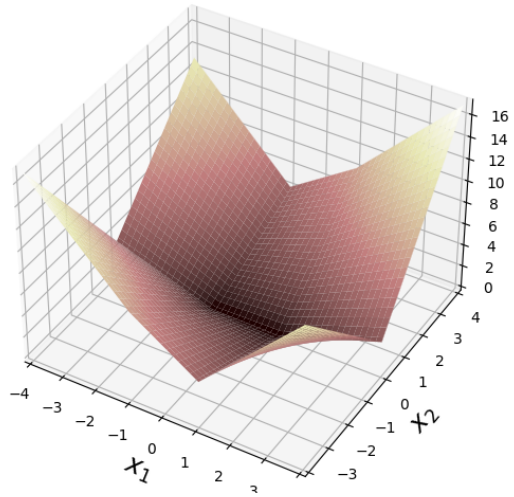

Fig. 1. An example of a function $f_1(x_1, x_2)$ as given in Thm. 2 for an $\mathcal{R} = [-a, a] \times [-b, b]$ with arbitrary $a > 1$ and $b > 0$.

1) *For every* $x_2 \in \mathcal{R}_2$, *the function* $f_1(\cdot, x_2) : \mathcal{R}_1 \to \mathbb{R}$ *is a computable continuous function that is convex and piecewise linear (with 3 linear pieces).*
2) *For every* $x_1 \in \mathcal{R}_1$, *the function* $f_1(x_1, \cdot) : \mathcal{R}_2 \to \mathbb{R}$ *is a computable continuous function that is convex and piecewise linear (with 2 linear pieces).*
3) f_1 *has only global minima (i.e. no local minima that are not global) and the set of all global minima is convex.*
4) *For every* $x_2 \in \mathcal{R}_2$, $x_2 \neq 0$, *the function* $f(\cdot, x_2)$ *has only one global minimum.*
5) *Every* $G_1 \in \mathcal{A}_1(f_1)$ *is not (Banach–Mazur) computable.*

Remark: The proof of this theorem can be found in [21]. There even a concrete functions f_1 with the properties from Thm. 2 is constructed. This function, shown in Fig. 1, illustrating the Properties 1)-4) of these functions.

As a consequence of Thm. 2, one immediately obtains a negative answer to Question 2, namely that the first optimization step, that aims to find the local minimum with respect to the first coordinate of $\mathbf{x}$, cannot algorithmically be solved:

Corollary 3: *Let* $f_1 : \mathcal{R}_1 \times \mathcal{R}_2 \to \mathbb{R}$ *be the computable continuous function from Thm. 2. Then the optimization step*

$$x_1^{(k+1)} = \arg\min_{y \in \mathcal{R}_1} f_1(y, x_2^{(k)})\,, \qquad k \in \mathbb{N}\,.$$

cannot be solved algorithmically, i.e. there exists no Turing machine that is able to compute $x_1^{(k+1)}$ *on input* $x_2^{(k)}$.

V. Reachability of Global Minimizers

We still consider the optimization problem (4) on a computable rectangle $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ with $\mathcal{R}_1 = [-a, a]$ and $\mathcal{R}_2 = [-b, b]$ for some positive $a, b \in \mathbb{R}_{\rm c}$ and for a given function $f : \mathbb{R}^2 \to \mathbb{R}$. In this section we need to study the point sets $\big\{(G_1(x_2), x_2) \subset \mathbb{R}^2 : x_2 \in \mathcal{R}_2\big\}$ and

$\big\{(x_1, G_2(x_1)) \subset \mathbb{R}^2 : x_1 \in \mathcal{R}_1\big\}$ in some detail. By the definition of the assignment functions (cf. Def. 6), we have

$$G_1(x_2) \in \mathcal{MIN}_1(x_2) = \arg\min_{x_1 \in \mathcal{R}_1} f(x_1, x_2) \quad \text{and}$$
$$G_2(x_1) \in \mathcal{MIN}_2(x_1) = \arg\min_{x_2 \in \mathcal{R}_2} f(x_1, x_2)\,.$$

Let $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2) \in \mathcal{MIN}_{\mathcal{R}}(f)$ be a global minimizer of f. We want to study the behavior of the points $(G_1(x_2), x_2) \in \mathbb{R}^2$ as x_2 approaches $\widehat{x}_2$ and the behavior of $(x_1, G_2(x_1)) \in \mathbb{R}^2$ as x_1 approaches $\widehat{x}_1$. To this end, we define the sets

$$\mathcal{G}_1^+(\widehat{x}_2) = \left\{ \mathbf{x} = \lim_{\substack{x_2 \to \widehat{x}_2 \\ x_2 > \widehat{x}_2}} \big(G_1(x_2), x_2\big) \;:\; G_1 \in \mathcal{A}_1(f) \right\}$$

$$\mathcal{G}_1^-(\widehat{x}_2) = \left\{ \mathbf{x} = \lim_{\substack{x_2 \to \widehat{x}_2 \\ x_2 < \widehat{x}_2}} \big(G_1(x_2), x_2\big) \;:\; G_1 \in \mathcal{A}_1(f) \right\}$$

i.e. the set of all limits of points of $(G_1(x_2), x_2) \in \mathbb{R}^2$ as x_2 converges to $\widehat{x}_2$ from above and below, respectively. In the same way, we define sets $\mathcal{G}_2^+(\widehat{x}_1)$ and $\mathcal{G}_2^-(\widehat{x}_1)$ as the limits of points $(x_1, G_2(x_1)) \in \mathbb{R}^2$ as x_1 converges to $\widehat{x}_1$ from above and below, respectively. By these definitions, we have

$$\mathcal{G}_1^{\pm}(\widehat{x}_2) \subset \mathcal{MIN}_{\mathcal{R}}(f) \quad \text{and} \quad \mathcal{G}_2^{\pm}(\widehat{x}_1) \subset \mathcal{MIN}_{\mathcal{R}}(f)$$

and we notice that these inclusions are generally strict. This motivates the following definition:

Definition 7: *Let $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2) \in \mathcal{MIN}_{\mathcal{R}}(f)$ be an arbitrary global minimizer of f. We say that $\widehat{\mathbf{x}}$ is reachable along*

- *the coordinate x_2 if $\widehat{\mathbf{x}} \in \mathcal{G}_1^+(\widehat{x}_2) \cup \mathcal{G}_1^-(\widehat{x}_2)$.*
- *the coordinate x_1 if $\widehat{\mathbf{x}} \in \mathcal{G}_2^+(\widehat{x}_1) \cup \mathcal{G}_2^-(\widehat{x}_1)$.*

Remark: In other words $\widehat{\mathbf{x}} \in (\widehat{x}_1, \widehat{x}_2)$ is reachable along the coordinate x_2 if there is a $G_1 \in \mathcal{A}_1(f)$ such that

$$\lim_{x_2 \to \widehat{x}_2,\, x_2 \neq \widehat{x}_2} \big(G_1(x_2), x_2\big) \in \mathcal{MIN}_{\mathcal{R}}(f)\,.$$

Example 1: Consider the function f_1 shown in Fig. 1. Its set of all global minimizers is $\mathcal{MIN}_{\mathcal{R}}(f_1) = \{(x_1, 0) : x_1 \in [-1, 1]\}$.
For all $x_2 > 0$, we have $(G_1(x_2), x_2) = (-1, x_2)$ and for all $x_2 < 0$, we have $(G_1(x_2), x_2) = (1, x_2)$, so that $\mathcal{G}_1^+(0) = (-1, 0)$ and $\mathcal{G}_1^-(0) = (1, 0)$. Similarly, since $(x_1, G_2(x_1)) = (x_1, 0)$ for all $x_1 \in \mathbb{R}$, we have $\mathcal{G}_2^+(x_1) = \mathcal{G}_2^-(x_1) = (x_1, 0)$ for every $x_1 \in [-1, 1]$.

Thus all points in $\mathcal{MIN}_{\mathcal{R}}(f_1)$ are reachable along the coordinate x_1 but only the points $(-1, 0)$ and $(1, 0)$ in $\mathcal{MIN}_{\mathcal{R}}(f_1)$ are reachable along the coordinate x_2.

If a minimizer $\widehat{\mathbf{x}} \in \mathcal{MIN}_{\mathcal{R}}(f)$ is not reachable along a certain coordinate then the iterative coordinate-wise algorithm will not be able to compute this minimizer. In such a case it might happen that even though f has global minimizers that are computable, the iterative coordinate-wise algorithm may not be able to compute them because they are not reachable along a certain coordinate.

The following theorem shows that there exist functions f_2 such that all global optimizers of f_2 that are reachable along the coordinate x_2 are not computable in $\mathbb{R}^2$.

Theorem 4: *Let $a, b \in \mathbb{R}_{\mathrm{c}}$ with $a, b > 0$ be arbitrary and let $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ with $\mathcal{R}_1 = [-a, a]$, $\mathcal{R}_2 = [-b, b]$. There exist computable continuous functions $f_2 : \mathcal{R} \to \mathbb{C}$ such that*

1) *$f_2 \in \mathcal{C}^1(\mathbb{R}^2)$.*
2) *$f_2(\cdot, x_2)$ is strictly convex for every $x_2 \in \mathbb{R}$, $x_2 \neq 0$, and a computable continuous function for every $x_2 \in \mathbb{R}_{\mathrm{c}}$.*
3) *$f_2(x_1, \cdot)$ is strictly convex for every $x_1 \in \mathbb{R}$, and a computable continuous function for every $x_1 \in \mathbb{R}_{\mathrm{c}}$.*
4) *f_2 has only global optimizers and the set of all global optimizers is a closed interval on the x_1-axis.*
5) *Every $(\widehat{x}_1, \widehat{x}_2) \in \mathcal{MIN}_{\mathcal{R}}(f_2)$ that can be reached along the coordinate x_2 is not computable in $\mathbb{R}^2$.*

Remark: The proof of this theorem can be found in [21] which also contains an explicit expression for a function f_2 with the properties given in Thm. 4.

As an immediate consequence of the last statement of this theorem we obtain the following negative answer to Question 1.

Corollary 5: *Let $f_2 : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a function from Thm. 4 and let $\{(x_n, \widehat{x}_2)\}_{n \in \mathbb{N}}$ be an arbitrary sequence that converges to a global minimum $(\widehat{x}_1, \widehat{x}_2) \in \mathcal{MIN}_{\mathcal{R}}(f_2)$ of f_2, then this convergence cannot be effective.*

Remark: In Thm. 4, f_2 belongs to $\mathcal{C}^1(\mathbb{R}^2)$. One can show that for every $K \in \mathbb{N}$, there exist functions $f_2 \in \mathcal{C}^K(\mathbb{R})$ such that all partial derivatives up to order K are computable continuous functions and satisfy Properties 2) - 5) of Thm. 4.

VI. Summary and Discussion

This paper has investigated the computability properties of iterative, block coordinate optimization algorithms. These results are relevant to situations where a global optimization is hard or impossible to implement. An example is a decentralized communication system where it is impossible to collect global information about the whole network at a central point, but where the optimization has to be performed locally, based on restricted knowledge on the network.

We have shown that there exist computable continuous functions f_*, for which it is impossible to find an effective implementation of such iterative optimization algorithms. Since it is impossible to construct an algorithm for single functions f_*, it is *a fortiori* impossible to construct a general algorithm who takes f as an input and which is able solve the optimization problem for a larger set (including the f_*'s) of functions.

We emphasis that the negative results of Theorems 2 and 4 are consequences of the local coordinate-wise optimization strategy. Indeed, the functions from Thm 2 and 4 have at least one global minimizer $\widehat{\mathbf{x}} \in \mathbb{R}^m$ that is computable. Therefore it is always possible to find a computable sequence $\big\{\widehat{\mathbf{x}}^{(k)}\big\}_{k \in \mathbb{N}} \subset \mathbb{R}^m$ that effectively converges to $\widehat{\mathbf{x}}$.

References

[1] K. Benidis, Y. Feng, and D. P. Palomar, "Optimization methods for financial index tracking: From theory to practice," *Foundations and Trends in Signal Processing*, vol. 3, no. 3, pp. 171–279, 2018.

[2] D. P. Palomar and Y. C. Eldar, Eds., *Convex Optimization in Signal Processing and Communications*. Cambridge, UK: Cambridge University Press, 2009.

[3] Y.-F. Liu, T.-H. Chang, M. Hong, Z. Wu, A. M.-C. So, E. A. Jorswieck, and W. Yu, "A survey of recent advances in optimization methods for wireless communications," pre-print, Jan. 2024, arXiv:2401.12025.

[4] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[5] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, Dec. 2009.

[6] Y. Lee, H. Boche, and G. Kutyniok, "Computability of optimizers for AI and data science," in *Handbook of Numerical Analysis*. New York: Elsevier B.V., 2024, pp. 1–54.

[7] W. Rhee, W. Yu, and J. M. Cioffi, "The optimality of beamforming in uplink multiuser wireless systems," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 86–96, Jan. 2004.

[8] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual SINR constraints," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18–28, Jan. 2004.

[9] E. Castañeda, A. Silva, A. Gameiro, and M. Kountouris, "An overview on resource allocation techniques for multi-user MIMO systems," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 1, pp. 239–284, Oct. 2017.

[10] L.-U. Choi and R. D. Murch, "A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 20–24, Jan. 2004.

[11] S. Shi, M. Schubert, and H. Boche, "Rate optimization for multiuser MIMO systems with linear processing," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 4020–4030, Aug. 2008.

[12] Y. Liu, L. Zhang, and M. A. Imran, "Multi-user beamforming and transmission based on intelligent reflecting surface," *IEEE Trans. Wireess Commun.*, vol. 21, no. 9, pp. 7329–7342, Sep. 2022.

[13] C. Meng, Z. Wei, D. Ma, W. Ni, L. Su, and Z. Feng, "Multiobjective-optimization-based transmit beamforming for multitarget and multiuser MIMO-ISAC systems," *IEEE Internet Things J.*, vol. 11, no. 18, pp. 29 260–29 274, Sep. 2024.

[14] C. Ding, C. Zeng, N. Su, J.-B. Wang, and M. Lin, "Transmit precoding for MIMO radar and MU-MIMO communication with ISAC," in *IEEE Global Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, 2023, pp. 1308–1313.

[15] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, Jul. 1972.

[16] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 14–20, Jan. 1972.

[17] I. Csiszár, "On the computation of rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 20, no. 1, pp. 122–124, Jan. 1974.

[18] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proc. London Math. Soc.*, vol. s2-42, no. 1, pp. 230–265, 1937.

[19] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem. A correction," *Proc. London Math. Soc.*, vol. s2-43, no. 1, pp. 544–546, 1938.

[20] M. B. Pour-El and J. I. Richards, *Computability in Analysis and Physics*. Berlin: Springer-Verlag, 1989.

[21] H. Boche, V. Pohl, and H. V. Poor, "Iterative optimization of multidimensional functions on Turing machines under performance guarantees," pre-print, Jan. 2025, arXiv:2501.13038.

[22] K. Weihrauch, *Computable Analysis*. Berlin: Springer-Verlag, 2000.

[23] H. M. Friedman, "The computational complexity of maximization and integration," *Adv. Math.*, vol. 53, no. 1, pp. 80–98, Jul. 1984.

[24] K.-I. Ko, *Complexity Theory of Real Functions*. Basel: Birkhäuser, 1991.

[25] J. Avigad and V. Brattka, "Computability and analysis: The legacy of Alan Turing," in *Turing's Legacy: Developments from Turing's Ideas in Logic*, ser. Lecture Notes in Logic, Bd. 42. New York: Cambridge University Press, 2014, pp. 1–47.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 20004.

[27] D. P. Bertsekas, *Convex Optimization Algorithms*. Nashua, USA: Athena Scientific, 2015.

[28] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, ser. MPS-SIAM Series on Optimization. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (SIAM), 2001.

[29] E. Specker, "Der Satz vom Maximum in der rekursiven Analysis," in *Ernst Specker Selecta*, G. Jäger, H. Läuchli, B. Scarpellini, and V. Strassen, Eds. Basel: Birkhäuser, 1990, pp. 148–159.

[30] M. J. D. Powell, "On search directions for minimization algorithms," *Math. Program.*, vol. 4, pp. 193–201, 1973.

[31] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, pp. 3–34, Jun. 2015.

[32] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM J. Optim.*, vol. 22, no. 2, pp. 341–362, 2012.

[33] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Oper. Res. Lett.*, vol. 26, no. 3, pp. 127–136, Apr. 2000.

[34] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.

[35] H. Boche, R. F. Schaefer, and H. V. Poor, "Algorithmic computability and approximability of capacity-achieving input distributions," *IEEE Trans. Inf. Theory*, vol. 69, no. 9, pp. 5449–5462, Sep. 2023.

[36] Y. Lee, H. Boche, and G. Kutyniok, "Computability of optimizers," *IEEE Trans. Inf. Theory*, vol. 70, no. 4, pp. 2967–2983, Apr. 2024.