Appl. Set-Valued Anal. Optim. 5 (2023), No. 3, pp. 325-346
Available online at http://asvao.biemdas.com
https://doi.org/10.23952/asvao.5.2023.3.02

SUPERIORIZATION: THE ASYMMETRIC ROLES OF FEASIBILITY-SEEKING AND OBJECTIVE FUNCTION REDUCTION

YAIR CENSOR

Department of Mathematics, University of Haifa, Mt. Carmel, Haifa 3498838, Israel

Abstract. The superiorization methodology can be thought of as lying conceptually between feasibility-seeking and constrained minimization. It is not trying to solve the full-fledged constrained minimization problem composed from the modeling constraints and the chosen objective function. Rather, the task is to find a feasible point which is "superior" (in a well-defined manner) with respect to the objective function, to one returned by a feasibility-seeking only algorithm. We telegraphically review the superiorization methodology and where it stands today and propose a rigorous formulation of its, yet only partially resolved, guarantee problem. The real-world situation in an application field is commonly represented by constraints defined by the modeling process and the data, obtained from measurements or otherwise dictated by the model-user. The feasibility-seeking problem requires to find a point in the intersection of all constraints without using any objective function to aim at any specific feasible point. At the heart of the superiorization methodology lies the modeler desire to use an objective function, that is exogenous to the constraints, in order to seek a feasible solution that will have lower (not necessarily minimal) objective function value. This aim is less demanding than full-fledged constrained minimization but more demanding than plain feasibility-seeking. Putting emphasis on the need to satisfy the constraints, because they represent the real-world situation, one recognizes the "asymmetric roles of feasibility-seeking and objective function reduction", namely, that fulfilling the constraints is the main task while reduction of the exogenous objective function plays only a secondary role. There are two research directions in the superiorization methodology that nourish from this same general principle: Weak superiorization and strong superiorization. Since its inception in 2007, the superiorization methodology has evolved and gained ground, as can be seen from the, compiled and continuously updated, bibliography at: http://math.haifa.ac.il/yair/bib-superiorization-censor.html.

Keywords. Bounded perturbation resilience; Cconstraints modeling; Dynamic string-averaging projections; Derivative-free; Proximity-target curve; Superiorization.

CONTENTS

Email address: yair@math.haifa.ac.il

Received October 24, 2022; Accepted December 29, 2022.

1. Introduction

This paper is a tutorial/review but it does not follow the classical sense of these notions. It mostly rather collects pieces from the published literature creating a collage[1]. This collage brings together definitions and results about the **superiorization methodology**, directs the reader to the existing literature, says a word about the history, and proposes a rigorous formulation of its, yet only partially resolved, guarantee problem.

It is a tribute to my friend, colleague and collaborator Simeon Reich whose papers (with Butnariu and Zaslavski) [12, 13] paved the way for the superiorization methodology, see Section 4 below, on the occasion of his 75th birthday.

The fundamental underlying question considered here is what to do with **input** that consists of a **constraints set** and an **objective function**? Do constrained minimization or do superiorization?

There are many available routes toward doing constrained minimization. These include, speaking generally, algorithms based on the use of Lagrange multipliers or on the regularization approach. In the latter approach, a **proximity function** that measures constraints violation is appended to the objective function, along with a regularization parameter, and algorithms for unconstrained minimization are applied to the resulting “regularized objective function”.

Other methods, like penalty methods and barrier methods, require a feasible initialization point (i.e., a point in the constraints set) and perform searches for the constrained minimum within the constraints set by “preventing” the iterates of these searches from “leaving” the constraints set.

Recognizing that the two pieces of the input (the constraints set and the objective function) are independent, there is the approach of handling the constrained minimization problem as an unconstrained bi-objective (i.e., multi-objective with two objectives) problem for the original objective function and for a proximity function.

In the regularization approach, mentioned above, the user decides how much importance should be attached to fulfilling the constraints versus minimizing the original objective function by defining the, so called, **regularization parameter**, which determines how much weight is given to each of the two tasks.

The superiorization methodology (SM) offers a different approach than those mentioned above. Putting emphasis on the need to satisfy the constraints, since they represent the real-world situation, the SM recognizes the asymmetric roles of feasibility-seeking and objective function

[1] As such, it inevitably includes some portions from previous publications which are all clearly acknowledged.

reduction. Namely, that the feasibility-seeking to fulfill the constraints, is the main task while reduction of the exogenous objective function plays only a secondary role.

In a nutshell, the SM does this by taking a convergent feasibility-seeking algorithm and perturbing its iterates so that the objective function values get reduced while retaining the overall convergence to a feasible point. The question whether such "local" objective function reduction steps, that are interlaced into the feasibility-seeking algorithm, accumulate toward a global objective function reduction, not necessarily minimization, is the gist of the yet not fully solved guarantee problem of the SM.

In the SM one takes an iterative algorithm, investigates its **perturbation resilience**, and then, uses proactively such permitted perturbations, to force the perturbed algorithm to do something useful in addition to what it is originally designed to do. The original unperturbed algorithm is called the **basic algorithm** and the perturbed algorithm is called the **superiorized version of the basic algorithm**.

If the basic algorithm is computationally efficient and useful in terms of the application at hand, and if the perturbations are simple and not expensive to calculate, then the advantage of this methodology is that, for essentially the computational cost of the basic algorithm, we are able to get something more by steering its iterates according to the perturbations.

This is a very general principle, which has been successfully used in a variety of important practical applications and awaits to be implemented and tested in additional fields; see the recent papers in the, compiled and continuously updated, bibliography of scientific publications on the superiorization methodology and perturbation resilience of algorithms [15].

Although not limited to this case, an important special case of the superiorization methodology is when the basic algorithm is a feasibility-seeking algorithm for a family of constraints and the perturbations that are interlaced into it aim at reducing, not necessarily minimizing, a given objective function. This case is what drives the intuition behind the superiorization methodology and is the subject matter of this paper[2].

Superiorization has a world-view that is quite different from that of classical constrained optimization. Both in superiorization and in classical constrained optimization there is an assumed domain $\Omega \subseteq R^n$ in the n-dimensional Euclidean[3] space and an exogenous objective (a.k.a. merit, criterion, target[4], etc.) function $\phi : R^n \to R$ that maps Ω into R. In classical optimization it is assumed that there is a constraints set $C \subset \Omega$ and the task is to find an $x \in C$ for which $\phi(x)$ is minimal over C. In superiorization the task is different: to find a feasible point in C which is "superior" (in a well-defined manner) with respect to the objective function, to one returned by a feasibility-seeking only algorithm.

There are two research directions in the superiorization methodology that nourish from the same general principle. One is the direction when the constraints are assumed to be consistent,

[2] Support for this reasoning may be borrowed from the American scientist and Noble-laureate Herbert Simon who was in favor of "satisficing" rather than "maximizing". Satisficing is a decision-making strategy that aims for a satisfactory or adequate result, rather than the optimal solution. This is because aiming for the optimal solution may necessitate needless expenditure of time, energy and resources. The term "satisfice" was coined by Herbert Simon in 1956 [72], see also: https://en.wikipedia.org/wiki/Satisficing.

[3] We limit most of our discussion to the finite-dimensional Euclidean space in order to present as clearly as possible our points although the notions and ideas can be carried over to Hilbert space or other infinite-dimensional spaces.

[4] We use "objective function" and "target function" interchangeably in the sequel.

$C \neq \emptyset$ and the notion of **bounded perturbation resilience** is used. In this case one treats the superiorized version of the basic algorithm as a recursion formula, without a stopping rule, that produces an infinite sequence of iterates and asymptotic convergence questions are in the focus of study.

The second direction does not assume consistency of the constraints but uses instead a **proximity function** that measures the violation of the constraints. Instead of seeking asymptotic feasibility, it looks at **ε-compatibility** and uses the notion of **strong perturbation resilience**. The same core superiorized version of the basic algorithm might be investigated in each of these directions, but the second is apparently more practical since it relates better to problems formulated and treated in practice. We use the terms **weak superiorization** and **strong superiorization** as a nomenclature for the first and second directions, respectively[5].

2. Constraints Oriented Modeling

Let R^n be the n-dimensional Euclidean space where $x \in R^n$ is represented by its components $x = (x_j)_{j=1}^n$. A **constraint** is a condition that restricts a vector x to belong to given sets that are represented by functional inequalities. **Modeling** is the process of representing a real-world problem, in some field of application, in a mathematical language amenable to mathematical analysis and to the development of tractable algorithmic solutions.

Constraints oriented modeling is a modeling process that represents the real-world problem by a system of constraints. The "individual" constraints $C_i \subset R^n$, for all $i = 1, 2, \ldots, m$, are

$$C_i := \{x \in R^n \mid q_i(x) \leq \gamma_i\},$$

where the function $q_i : R^n \to R$ is a mapping into the reals and $\gamma_i \in R$ and C_i is the γ_i-level set of q_i. This gives rise to the **feasibility-seeking problem**

$$\text{find } x \in C := \cap_{i=1}^m C_i. \tag{2.1}$$

We assume that this feasibility-seeking problem is **consistent**, i.e., $C \neq \emptyset$. In the inconsistent case the statement $x \in C$ is meaningless and alternative solution concepts must be used, but we do not wander in this direction here, see, e.g., [28] and references therein.

The word "find" in (2.1) is meant as either actually presenting such an $x \in C$ or generating an infinite sequence $\{x^k\}_{k=0}^\infty$, with $x^k \in R^n$ for all $k \geq 0$, that asymptotically converges such that $\lim_{k\to\infty} x^k = x^* \in C$. In strong superiorization, however, "find" has another meaning, see Section 4.2 below.

A particular instance is when all constraints are convex sets represented as level sets of convex functions. In that case (2.1) is the well-known **convex feasibility problem** (CFP), consult [7] and [14, Subsection 1.3.4], which is, in turn, a special instance of the **common fixed point problem** (CFPP) of a family of operators, consult [8, Subsection 4.6].

In the constraints oriented modeling process the constraints functions $\{q_i\}_{i=1}^m$ are usually decided upon by the modeler to best represent the nature of the problem that is modeled. The **level sets parameters** $\{\gamma_i\}_{i=1}^m$ are commonly obtained through physical measurements or via prescriptions defined by the user. Thus, $\{q_i\}_{i=1}^m$ and $\{\gamma_i\}_{i=1}^m$ are the **input** for the feasibility-seeking problem.

[5] These terms were proposed in [31], following a private discussion with our colleague and coworker in this field Gabor Herman.

A question that might arise at this point is why not translate the feasibility-seeking problem upfront to an unconstrained minimization of a proximity function that measures the violation of the constraints or by using the indicator functions of the individual sets? Or why not translate the feasibility-seeking problem upfront to a constrained minimization of an objective function which is a constant over the constraints at hand? While these approaches are legitimate they do not necessarily lead to the same "algorithmic territory". It is not clear, and to our understanding doubtful, whether better algorithms for the feasibility-seeking problem can be discovered when using such translational approaches.

On the contrary, quite a few well-known algorithms for feasibility-seeking were, and still are, discovered, developed and studied by using tools from outside mathematical optimization. for example, the feasibility-seeking problem can be formulated as a special case of the common fixed point problem, mentioned above, for the case when the operators are projections, and as such benefit from the large body of knowledge in the field of fixed point theory. The excellent book of Cegielski [14] attests to this line of work.

3. Feasibility-Seeking Algorithms

We look at the **convex feasibility problem** (CFP) which is to find a feasible point $x^* \in C$ when all sets C_i are convex and commonly also closed. This prototypical problem underlies the modeling of a variety of real-world problems in many fields, see, e.g., the pointers and references in Bauschke and Borwein [7, Section 1] and in Cegielski's book [14, Section 1.3].

If $C \neq \emptyset$ does not hold then the CFP is inconsistent and a feasible point does not exist, see, e.g., the review of inconsistent feasibility problems [28]. However, algorithmic research of inconsistent CFPs exists and is mainly focused on two directions. One is oriented toward defining solution concepts other than $x^* \in C$ that will apply, such as proximity function minimization wherein a **proximity function** measures in some way the total violation of all constraints, see, e.g., [14, pp. 28-29.]. The second direction investigates the behavior of algorithms that are designed to solve a consistent CFP when applied to inconsistent problems. The latter is fueled by situations wherein one lacks a priori information about the consistency or inconsistency of the CFP or does not wish to invest computational resources to get hold of such knowledge prior to running his algorithm.

Projection methods. Projections onto sets are used in a wide variety of methods in optimization theory but not every method that uses projections really belongs to the class of projection methods as we mean it here. Here **projection methods** are iterative algorithms[6] that use projections onto sets while relying on the general principle that when a family of (usually closed and convex) sets is present then projections (or approximate projections) onto the given individual sets are easier to perform than projections onto other sets (intersections, image sets under some transformation, etc.) that are derived from the family of individual sets.

A projection algorithm reaches its goal, related to the whole family of sets, by performing projections onto the individual sets. Projection algorithms employ projections (or approximate projections) onto convex sets in various ways. They may use different kinds of projections, e.g.,

[6] As common, we use the terms algorithm or algorithmic structure for the iterative processes studied here although no termination criteria, which are by definition necessary in an algorithm, are present and only the asymptotic behavior of these processes is studied. This does not create any ambiguity because whether we consider an infinite iterative process or an algorithm with a termination rule is always clear from the context.

orthogonal (least Euclidean distance) projections, Bregman projections, entropy projections, subgradient projections, intrepid projections, valiant projections, Douglas-Rachford operators etc. and, sometimes, even use different projections within the same algorithm. They serve to solve a variety of problems which are either of the feasibility-seeking or the constrained optimization types. They have different algorithmic structures, of which some are particularly suitable for parallel computing, and they demonstrate nice convergence properties and/or good initial behavior patterns in some significant fields of applications.

Apart from theoretical interest, the main advantage of projection methods, which makes them successful in real-world applications, is computational. They commonly have the ability to handle huge-size problems of dimensions beyond which other, more sophisticated currently available, methods cease to be efficient. This is so because the building bricks of a projection algorithm are the projections onto the individual sets (assumed and actually easy to perform) and the algorithmic structures are either sequential or simultaneous or in-between, such as in the **block-iterative projection** (BIP) methods, see, e.g., [33, 61, 64], or in the more recent **string-averaging projection** (SAP) methods, see details and references in Subsection 4.1. An advantage of projection methods is that they work with initial data and do not require transformation of, or other operations on, the sets describing the problem.

4. The Superiorization Methodology

Since its inception in 2007, the superiorization method has evolved and gained ground. Quoting and distilling from earlier publications, we review here the two directions of the superiorization methodology. Recent review papers on the subject which could be read together with this paper are Herman's [50] and [51]. Unless otherwise stated, we restrict ourselves, for simplicity, to the n-dimensional Euclidean space R^n although some materials below remain valid in Hilbert space.

Recent publications on the superiorization methodology (SM) are devoted to either weak or strong superiorization, without yet using these terms. They are [9, 11, 18, 29, 39, 40, 43, 52, 53, 55, 64, 68]. many of the papers contain a detailed description of the SM, its motivation, and an up-to-date review of SM-related previous work.

The superiorization method was born when the terms and notions "superiorization" and "perturbation resilience", in the present context, first appeared in the 2009 paper [39] which followed its 2007 forerunner by Butnariu et al. [11]. The ideas have some of their roots in the 2006 and 2008 papers of Butnariu et al. [12, 13] where it was shown that if iterates of a nonexpansive operator converge for any initial point, then its inexact iterates with summable errors also converge.

Bounded perturbation resilience of a parallel projection method was observed as early as 2001 in [34, Theorem 2] (without using this term). All these culminated in Ran Davidi's 2010 PhD dissertation [38] and the many papers that appeared since then and are cited in [15]. The latter is a Webpage dedicated to superiorization and perturbation resilience of algorithms that contains a continuously updated bibliography on the subject. This Webpage[7] is source for the wealth of work done in this field to date, including two special issues of journals [22] and [45] dedicated to research of the SM. Recent work includes [10, 48, 54, 63, 75]. Interestingly, [1] notices some structural similarities of the SM with incremental proximal gradient methods.

[7] http://math.haifa.ac.il/yair/bib-superiorization-censor.html#top, last updated on August 17, 2023 with 177 items.

Let T denote a mathematically-formulated problem, of any kind or sort, with solution set Ψ_T. The following cases immediately come to mind although any T and its Ψ_T can potentially be used.

Case 4.1. *T is a convex feasibility problem (CFP) of the form: find a vector $x^* \in \cap_{i=1}^m C_i$, where $C_i \subseteq R^n$ are closed convex subsets. In this case $\Psi_T = \cap_{i=1}^m C_i$.*

Case 4.2. *T is a constrained minimization problem:* minimize$\{f(x) \mid x \in \Phi\}$ *of an objective function f over a feasible region Φ. In this case $\Psi_T = \{x^* \in \Phi \mid f(x^*) \leq f(x) \text{ for all } x \in \Phi\}$.*

The superiorization methodology is intended for function reduction problems of the following form.

Problem 4.1. The Function Reduction Problem. Let $\Psi_T \subseteq R^n$ be the solution set of some given mathematically-formulated problem T and let $\phi : R^n \to R$ be an objective function. Let $\mathscr{A} : R^n \to R^n$ be an algorithmic operator that defines an iterative **basic algorithm** for the solution of T. Find a vector $x^* \in \Psi_T$ whose function ϕ value is smaller or equal than that of a point in Ψ_T that would have been reached by applying the Basic Algorithm for the solution of problem T.

As explained below, the superiorization methodology approaches this problem by automatically generating from the basic algorithm its **superiorized version of the basic algorithm**. The vector x^* obtained from the superiorized version of the basic algorithm need not be a minimizer of ϕ over Ψ_T. Another point to observe is that the very problem formulation depends not only on the data T, Ψ_T and ϕ but also on the pair of algorithms – the original unperturbed basic algorithm, represented by $\mathscr{A}$, for the solution of problem T, and its superiorized version.

A fundamental difference between weak and strong superiorization lies in the meaning attached to term "solution of problem T" in Problem 4.1. In weak superiorization solving the problem T is understood as generating an infinite sequence $\{x^k\}_{k=0}^\infty$ that converges to a point $x^* \in \Psi_T$, thus, Ψ_T must be nonempty. In strong superiorization solving the problem T is understood as finding a point x^* that is **ε-compatible with** Ψ_T, for some positive ε, thus, nonemptiness of Ψ_T need not be assumed.

We concentrate in the next sections mainly on Case 4.1. Superiorization work on Case 4.2, e.g., where T is a maximum likelihood optimization problem and Ψ_T – its solution set, appears in [43, 55, 59].

4.1. **Weak superiorization.** In weak superiorization the set Ψ_T is assumed to be nonempty and one treats the "Superiorized Version of the Basic Algorithm" as a recursion formula that produces an infinite sequence of iterates. The SM strives to asymptotically find a point in Ψ_T which is superior, i.e., has a lower, but not necessarily minimal, value of the ϕ function, to one returned by the Basic Algorithm that solves the original problem T only.

This is done by first investigating the bounded perturbation resilience of an available Basic Algorithm designed to solve efficiently the original problem T and then proactively using such permitted perturbations to steer its iterates toward lower values of the ϕ objective function while not loosing the overall convergence to a point in Ψ_T.

Definition 4.1. Bounded perturbation resilience (BPR). Let $\Gamma \subseteq R^n$ be a given nonempty set. An algorithmic operator $\mathscr{A} : R^n \to R^n$ is said to be **bounded perturbations resilient with respect to** Γ if the following is true: If a sequence $\{x^k\}_{k=0}^\infty$, generated by the iterative process

$x^{k+1} = \mathscr{A}(x^k)$, for all $k \geq 0$, converges to a point in Γ for all $x^0 \in R^n$, then any sequence $\{y^k\}_{k=0}^{\infty}$ of points in R^n that is generated by $y^{k+1} = \mathscr{A}(y^k + \beta_k v^k)$, for all $k \geq 0$, also converges to a point in Γ for all $y^0 \in R^n$ provided that, for all $k \geq 0$, $\beta_k v^k$ are **bounded perturbations**, meaning that $\beta_k \geq 0$ for all $k \geq 0$ such that $\sum_{k=0}^{\infty} \beta_k < \infty$, and that the sequence $\{v^k\}_{k=0}^{\infty}$ is bounded.

Let $\phi : R^n \to R$ be a real-valued convex continuous function and let $\partial\phi(z)$ be the subgradient set of ϕ at z and, for simplicity of presentation, assume here that $\Gamma = R^n$. In other specific cases care must be taken regarding how Γ and Ψ_T are related. The following superiorized version of the basic algorithm $\mathscr{A}$ is based on [31, Algorithm 4.1].

Algorithm 1 Superiorized version of the basic algorithm $\mathscr{A}$

```
Initialization: Let $N$ be a natural number, and let $y^0 \in R^n$ be an arbitrary user-chosen vector.
Iterative Step: Given a current iteration vector $y^k$, pick an $N_k \in \{1,2,\ldots,N\}$ and start an inner loop of calculations as follows:
Step 1. Inner loop initialization: Define $y^{k,0} = y^k$.
Step 2. Inner loop step: Given $y^{k,n}$, as long as $n < N_k$, do as follows:
  Step 2-1. Pick a $0 < \beta_{k,n} \leq 1$ in a way that guarantees that
      $\sum_{k=0}^{\infty} \sum_{n=0}^{N_k - 1} \beta_{k,n} < \infty.$
  Step 2-2. Pick an $s^{k,n} \in \partial\phi(y^{k,n})$ and define $v^{k,n}$ as follows:
      $v^{k,n} = \begin{cases} -\frac{s^{k,n}}{\left\| s^{k,n} \right\|}, & \text{if } 0 \notin \partial\phi(y^{k,n}), \\ 0, & \text{if } 0 \in \partial\phi(y^{k,n}). \end{cases}$
  Step 2-3. Calculate the perturbed iterate
      $y^{k,n+1} = y^{k,n} + \beta_{k,n} v^{k,n}$
and if $n+1 < N_k$ set $n \leftarrow n+1$ and go to Step 2, otherwise go to Step 3.
Step 3. Exit the inner loop with the vector $y^{k,N_k}$.
Step 4. Calculate
      $y^{k+1} = \mathscr{A}(y^{k,N_k})$
set $k \leftarrow k+1$ and go back to the beginning of the Iterative Step.
```

Considering Case 4.1 in Section 4 where T is a convex feasibility problem, the Dynamic String-Averaging Projection (DSAP) method of [29] that we describe here constitutes a family of algorithmic operators that can play the role of the above $\mathscr{A}$ in a basic algorithm for the solution of the CFP T.

Let $\{C_i\}_{i=1}^m$ be nonempty closed convex subsets of a Hilbert space X, set $C = \cap_{i=1}^m C_i$, and assume $C \neq \emptyset$. For $i = 1,2,\ldots,m$, denote by $P_i := P_{C_i}$ the orthogonal (least Euclidean distance) projection onto the set C_i. An **index vector** is a vector $t = (t_1,t_2,\ldots,t_q)$ such that $t_s \in \{1,2,\ldots,m\}$ for all $s = 1,2,\ldots,q$, whose length is $\ell(t) = q$. The product of the individual projections onto the sets whose indices appear in the index vector t is $P[t] := P_{t_q} \cdots P_{t_2} P_{t_1}$, called a **string operator**.

A finite set Ω of index vectors is called **fit** if for each $i \in \{1,2,\ldots,m\}$, there exists a vector $t = (t_1,t_2,\ldots,t_q) \in \Omega$ such that $t_s = i$ for some $s \in \{1,2,\ldots,q\}$. Denote by $\mathscr{M}$ the collection of all pairs (Ω,w), where Ω is a finite fit set of index vectors and $w : \Omega \to (0,\infty)$ is such that $\sum_{t\in\Omega} w(t) = 1$.

For any $(\Omega,w) \in \mathscr{M}$, define the convex combination of the end-points of all strings defined by members of Ω

$$P_{\Omega,w}(x) := \sum_{t\in\Omega} w(t)P[t](x),\ x \in X.$$

Let $\Delta \in (0,1/m)$ and an integer $\bar{q} \geq m$ be arbitrary fixed and denote by $\mathscr{M}_* \equiv \mathscr{M}_*(\Delta,\bar{q})$ the set of all $(\Omega,w) \in \mathscr{M}$ such that the lengths of the strings are bounded and the weights are all bounded away from zero, i.e.,

$$\mathscr{M}_* := \{(\Omega,w) \in \mathscr{M} \mid \ell(t) \leq \bar{q} \text{ and } w(t) \geq \Delta,\ \forall\, t \in \Omega\}. \tag{4.1}$$

Algorithm 2 The DSAP method with variable strings and variable weights

```
**Initialization:** Select an arbitrary $x^0 \in X$.
**Iterative Steps**: Given a current iteration vector $x^k$, pick a pair $(\Omega_k,w_k) \in \mathscr{M}_*$ and calculate
the next iteration vector $x^{k+1}$ by
    $x^{k+1} = P_{\Omega_k,w_k}(x^k).$
```

The first prototypical string-averaging algorithmic scheme appeared in [19] and subsequent work on its realization with various algorithmic operators includes [25, 26, 27, 30, 37, 46, 67, 68, 70], see also [5] and Zaslavski's book [76]. If in the DSAP method one uses only a single index vector $t = (1,2,\ldots,m)$ that includes all constraints indices then the fully-sequential Kaczmarz cyclic projection method is obtained. For linear hyperplanes as constraints sets the latter is equivalent with the, independently discovered, ART (Algebraic Reconstruction Technique) in image reconstruction from projections, see [49]. If, at the other extreme, one uses exactly m index vectors $t = (i)$, for $i = 1,2,\ldots,m$, each consisting of exactly one constraint index, then the fully-simultaneous projection method of Cimmino is recovered. In-between these “extremes” the DSAP method allows for a large arsenal of specific feasibility-seeking projection algorithms. See [7, 16, 17] for more information on projection methods.

The **superiorized version of the DSAP algorithm** is obtained by using Algorithm 2 as the algorithmic operator $\mathscr{A}$ in Algorithm 1. The following result about its behavior was proved. Consider the set $C_{min} := \{x \in C \mid\ \phi(x) \leq \phi(y) \text{ for all } y \in C\}$, and assume that $C_{min} \neq \emptyset$.

Theorem 4.1. [31, Theorem 4.1] *Let $\phi : X \to R$ be a convex continuous function, and let $C_* \subseteq C_{min}$ be a nonempty subset. Let $r_0 \in (0,1]$ and $\bar{L} \geq 1$ be such that, for all $x \in C_*$ and all y such that, $||x-y|| \leq r_0$, $|\phi(x)-\phi(y)| \leq \bar{L}||x-y||$, and suppose that $\{(\Omega_k,w_k)\}_{k=0}^{\infty} \subset \mathscr{M}_*$. Then any sequence $\{y^k\}_{k=0}^{\infty}$, generated by the superiorized version of the DSAP algorithm, converges in the norm of X to a $y^* \in C$ and exactly one of the following two alternatives holds:*

(a) $y^* \in C_{min}$;

(b) $y^* \notin C_{min}$ *and there exist a natural number* k_0 *and a* $c_0 \in (0,1)$ *such that for each* $x \in C_*$ *and for each integer* $k \geq k_0$*,*

$$\|y^{k+1} - x\|^2 \leq \|y^k - x\|^2 - c_0 \sum_{n=1}^{N_k - 1} \beta_{k,n}.$$

This demonstrates that $\{y^k\}_{k=0}^{\infty}$ is strictly Fejér-monotone with respect to C_*, i.e., that

$$\|y^{k+1} - x\|^2 < \|y^k - x\|^2,$$

for all $k \geq k_0$, because $c_0 \sum_{n=1}^{N_k-1} \beta_{k,n} > 0$. The strict Fejér-monotonicity however does not guarantee convergence to a constrained minimum point but only says that the so-created feasibility-seeking sequence $\{y^k\}_{k=0}^{\infty}$ has the additional property of getting strictly closer, without necessarily converging, to the points of a subset of the solution set of of the constrained minimization problem.

Published experimental results repeatedly confirm that global reduction of the value of the objective function ϕ is indeed achieved, without loosing the convergence toward feasibility, see [9, 11, 18, 29, 39, 40, 43, 52, 53, 55, 64, 68]. In some of these cases the SM returns a lower value of the objective function ϕ than an exact minimization method with which it is compared, e.g., [20].

4.2. **Strong superiorization.** In this section we present a restricted version of the SM of [53] as adapted to the situation in [20]. We consider again Case 4.1 in Section 4 wherein T is a convex feasibility problem. Let $\Theta := \{C_i\}_{i=1}^m$ be a family of nonempty closed convex subsets of a Hilbert space X and set $C = \cap_{i=1}^m C_i$. We do not assume that $C \neq \emptyset$, but only that there is some nonempty subset $\Lambda \in R^n$ such that $C_i \subseteq \Lambda$ for all i. Instead of the nonemptiness assumption we associate with the family $\{C_i\}_{i=1}^m$ a **proximity function** $Prox_\Theta : \Lambda \to \mathbb{R}_+$ that is an indicator of how incompatible an $x \in \Lambda$ is with the constraints. For any given $\varepsilon > 0$, a point $x \in \Lambda$ for which $Prox_\Theta(x) \leq \varepsilon$ is called an **ε-compatible solution for Θ**. We further assume that we have a feasibility-seeking algorithmic operator $\mathscr{A} : R^n \to \Lambda$, with which we define the basic algorithm as the iterative process

$$x^{k+1} = \mathscr{A}(x^k), \text{ for all } k \geq 0, \text{ for an arbitrary } x^0 \in \Lambda.$$

The following definition helps to evaluate the output of the basic algorithm upon termination by a stopping rule.

Definition 4.2. The ε-output of a sequence. Given Θ such that $C_i \subseteq \Lambda \subseteq R^n$, for all i, a proximity function $Prox_\Theta : \Lambda \to R_+$, a sequence $\left\{x^k\right\}_{k=0}^{\infty} \subset \Lambda$ and an $\varepsilon > 0$, then an element x^K of the sequence which has the properties: (i) $Prox_\Theta\left(x^K\right) \leq \varepsilon$, and (ii) $Prox_\Theta\left(x^k\right) > \varepsilon$ for all $0 \leq k < K$, is called an **ε-output of the sequence $\left\{x^k\right\}_{k=0}^{\infty}$ with respect to the pair** (Θ, $Prox_\Theta$)**.**

We denote the **ε-output** by $O\left(\Theta, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right) = x^K$. Clearly, an ε-output $O\left(\Theta, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right)$ of a sequence $\left\{x^k\right\}_{k=0}^{\infty}$ might or might not exist, but if it does, then it is unique. If $\left\{x^k\right\}_{k=0}^{\infty}$ is produced by an algorithm intended for the feasible set C, such as the Basic Algorithm, without a termination criterion, then $O\left(\Theta, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right)$ is the output produced by that algorithm when it includes the termination rule to stop when an ε-compatible solution for Θ is reached.

Definition 4.3. Strong perturbation resilience. Given Θ such that $C_i \subseteq \Lambda \subseteq R^n$, for all i, a proximity function $Prox_\Theta : \Lambda \to R_+$, an algorithmic operator $\mathscr{A}$ and an $x^0 \in \Lambda$. We use $\left\{x^k\right\}_{k=0}^{\infty}$ to denote the sequence generated by the Basic Algorithm when it is initialized by x^0. The basic algorithm is said to be **strongly perturbation resilient** iff the following hold: (i) there exist an $\varepsilon > 0$ such that the ε-output $O\left(\Theta, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right)$ exists for every $x^0 \in \Lambda$; (ii) for every $\varepsilon > 0$, for which the ε-output $O\left(\Theta, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right)$ exists for every $x^0 \in \Lambda$ it holds that the ε'-output $O\left(\Theta, \varepsilon', \left\{y^k\right\}_{k=0}^{\infty}\right)$ exists for every $\varepsilon' > \varepsilon$ and for every sequence $\left\{y^k\right\}_{k=0}^{\infty}$ generated by

$$y^{k+1} = \mathscr{A}\left(y^k + \beta_k v^k\right), \text{ for all } k \geq 0, \tag{4.2}$$

where the vector sequence $\left\{v^k\right\}_{k=0}^{\infty}$ is bounded and the scalars $\{\beta_k\}_{k=0}^{\infty}$ are such that $\beta_k \geq 0$, for all $k \geq 0$, and $\sum_{k=0}^{\infty} \beta_k < \infty$.

A theorem which gives sufficient conditions for strong perturbation resilience of the basic algorithm has been proved in [53, Theorem 1].

Along with Θ such that $C_i \subseteq \Lambda \subseteq R^n$, for all i and a proximity function $Prox_\Theta : \Lambda \to R_+$, we look at the objective function $\phi : R^n \to R$, with the convention that a point in R^n whose value of ϕ is smaller is considered **superior** to a point in R^n for which the value of ϕ is larger. The essential idea of the SM is to make use of the perturbations of (4.2) to transform a strongly perturbation resilient Basic Algorithm that seeks a constraints-compatible solution for Θ into its Superiorized Version whose outputs are equally good from the point of view of constraints-compatibility, but are superior (not necessarily optimal) according to the objective function ϕ.

Definition 4.4. Given a function $\phi : R^n \to R$ and a point $y \in R^n$, we say that a vector $d \in R^n$ is **nonascending for ϕ at** y iff $\|d\| \leq 1$ and there is a $\delta > 0$ such that for all $\lambda \in [0, \delta]$ we have $\phi\left(y + \lambda d\right) \leq \phi\left(y\right)$.

Obviously, the zero vector is always such a vector, but for superiorization to work we need a sharp inequality to occur in Definition 4.4 frequently enough. The Superiorized Version of the Basic Algorithm assumes that we have available a summable sequence $\{\eta_\ell\}_{\ell=0}^{\infty}$ of positive real numbers (for example, $\eta_\ell = a^\ell$, where $0 < a < 1$) and it generates, simultaneously with the sequence $\left\{y^k\right\}_{k=0}^{\infty}$ in Λ, sequences $\left\{v^k\right\}_{k=0}^{\infty}$ and $\{\beta_k\}_{k=0}^{\infty}$. The latter is generated as a subsequence of $\{\eta_\ell\}_{\ell=0}^{\infty}$, resulting in a nonnegative summable sequence $\{\beta_k\}_{k=0}^{\infty}$. The algorithm further depends on a specified initial point $y^0 \in \Lambda$ and on a positive integer N. It makes use of a logical variable called *loop*. The general form of the superiorized version of the basic algorithm is presented below by its pseudo-code.

Theorem 4.2. *Any sequence $\left\{y^k\right\}_{k=0}^{\infty}$, generated by the superiorized version of the basic algorithm, Algorithm 3, satisfies (4.2). Further, if, for a given $\varepsilon > 0$, the ε-output $O\left(\Theta, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right)$ of the Basic Algorithm exists for every $x^0 \in \Lambda$, then every sequence $\left\{y^k\right\}_{k=0}^{\infty}$, generated by the Algorithm 3, has an ε'-output $O\left(\Theta, \varepsilon', \left\{y^k\right\}_{k=0}^{\infty}\right)$ for every $\varepsilon' > \varepsilon$.*

The proof of this theorem follows from the analysis of the behavior of the superiorized version of the basic algorithm in [53, pp. 5537–5538]. In other words, Algorithm 3 produces outputs that are essentially as constraints-compatible as those produced by the original basic algorithm. However, due to the repeated steering of the process by lines 7 to 17 toward reducing the value

Algorithm 3 General form of the superiorized version of the basic algorithm

```
set $k = 0$
set $y^k = y^0$
set $\ell = -1$
repeat
    set $n = 0$
    set $y^{k,n} = y^k$
    while $n<N$
        set $v^{k,n}$ to be a nonascending vector for $\phi$ at $y^{k,n}$
        set $loop=true$
        while $loop$
            set $\ell = \ell + 1$
            set $\beta_{k,n} = \eta_\ell$
            set $z = y^{k,n} + \beta_{k,n} v^{k,n}$
            if $\phi(z) \leq \phi(y^k)$ then
                set $n = n+1$
                set $y^{k,n} = z$
                set $loop = false$
    set $y^{k+1} = \mathscr{A}(y^{k,N})$
    set $k = k+1$
```

of the objective function ϕ, we can expect that its output will be superior (from the point of view of ϕ) to the output of the (unperturbed) Basic Algorithm.

Algorithm 1 and Algorithm 3 are not identical but are based on the same leading principle of the superiorization methodology. Comments on the differences between them can be found in [31, Remark 4.1]. Nevertheless, the Theorems 4.1 and 4.2 related to these superiorized versions of the basic algorithm, respectively, leave the question of rigorously analyzing the behavior of the SM, under various conditions, open.

4.3. **Controlling the effect of the perturbations.** The scalars β_k in the SM algorithmic scheme, see Definition 4.1 and Eq. (4.2), are generated such that $\beta_k \geq 0$, for all $k \geq 0$, and $\sum_{k=0}^{\infty} \beta_k < \infty$. This implies that they form a tending to zero sequence. As step-sizes of the perturbation the effects of the objective function decrease is bound to diminish as iterations proceed.

In some applications, various methods have been studied for controlling the step-sizes, see, e.g., [57, 69], see also the software package SNARK14 [73] which is an updated version of [56]. Recently, a new strategy which allows restarting the sequence of step-sizes to a previous value while maintaining the summability of the series of step-sizes was suggested [2], resulting in improvement of the algorithm's performance.

5. Derivative-Free Superiorization

5.1. **Derivative-free superiorization and derivative-free optimization.** Here we describe the general applicability of **derivative-free superiorization** (DFS) as an alternative to previously proposed superiorization approaches. These earlier approaches were based on generation of nonascending vectors, for objective function reduction steps, that mostly required the ability to calculate gradients or subgradients of the objective function. Observing the body of knowledge of **derivative-free optimization** (DFO), see, e.g., [35], we explore a DFS algorithm.

In DFS, in the perturbation phase of the superiorized version of a basic algorithm the objective function reduction steps that depend on gradient or subgradient calculations are replaced by steps that use a direction search technique which does not require any form of differentiability. Continuing the work of [24], we searched in [21] the neighborhood of a current point x for a point at which the objective function exhibits nonascent.

While this might seem a simple technical matter, the ramifications for practical applications of the SM are important. For example, in intensity-modulated radiation therapy treatment planning, with photons, protons or other particles, the **normal tissue complication probability** (NTCP) is a predictor of radiobiological effects for organs at risk. The inclusion of it, or of other biological functions, as an objective function in the mathematical problem modeling and the planning algorithm, is hampered because they are, in general, empirical functions whose derivatives cannot be calculated, see, e.g., [44]. In the recent paper [65] the authors list issues of immediate clinical and practical relevance to the Proton Therapy community, highlighting the needs for the near future but also in a longer perspective. They say that "...practical tools to handle the variable biological efficiency in Proton Therapy are urgently demanded...".

The output of a superiorized version of a constraints-compatibility-seeking algorithm will have smaller (but not minimal) objective function ϕ value than the output of the same constraints-compatibility-seeking algorithm without perturbations, everything else being equal. Even though superiorization is not an exact minimization method, we think of it as an applicable (and possibly, more efficacious) alternative to derivative-free constrained minimization methods applied to the same data for two main reasons: its ability to handle constraints and its ability to cope with very large-size problems. This is in contrast with the current state of the art, which is as follows.

The review paper of Rios and Sahinidis [71] "... addresses the solution of *bound-constrained* optimization problems using algorithms that require only the availability of objective function values but no derivative information," with bound constraints imposed on the vector x. The book by Conn, Scheinberg and Vicente [35] deals only with derivative-free unconstrained minimization, except for its last chapter (of 10 pages out of the 275) entitled "Review of constrained and other extensions to derivative-free optimization." Li *et al*. [58] do not even mention constraints. In [41] the numerical work deals with: "The dimension of the problems [i.e., the size of the vector x] varies between 2 and 16, while the number of constraints are between 1 and 38, exceeding 10 in only 5 cases." In [32] the numerical tests are limited to: "The first case has 80 optimization variables [i.e., the size of the vector x] and only bound constraints, while the second example is a generally constrained production optimization involving 20 optimization variables and 5 general constraints." Similar orders of magnitude for problem sizes appear in the numerical results presented in [3] and also in the book of Audet and Hare [4].

This indicates that (i) much of the literature on derivative-free minimization is concerned with unconstrained minimization or with bound-constraints on the variables, and (ii) many, if not all, proposed methods were designed (or, at least, demonstrated) only for small-scale problems. In contrast, the DFS method proposed here can handle any type of constraints for which a separate efficient constraints-compatibility-seeking algorithm is available and is capable of solving very large problems. In the matter of problem sizes, we discover here, admittedly with a very preliminary demonstration, that DFS can compete well with DFO on large problems.

Since the constraints-compatibility-seeking algorithm forms part of the proposed DFS method, the method can use exterior initialization (that is initializing the iterations at any point in space). Furthermore, very large-scale problems can be accommodated.

The **progressive barrier** (PB) approach, described in Chapter 12 of the book [4], originally published in [3], is an alternative to the **exterior penalty** (EP) approach that is mentioned in [21]. However, the PB differs from our DFS method, in spite of some similarities with it, as explained in [21].

5.2. **The proximity-target curve.** A tool for deciding which of two iterative methods is "better" for solving a particular problem was presented in [21] and applied to the DFS algorithm developed there. We care to reproduce it here because of its potential usefulness in other situations where two algorithms are compared. Since an iterative method produces a sequence of points, our tool is based on such sequences. Furthermore, since in the SM we are interested in the values of two functions (proximity function and target function) at each iteration point, the efficacy of the behavior of the iterative method can be represented by a curve in two-dimensional space, defined as **the proximity-target curve** below.

It indicates the target value for any achieved proximity value. This leads to the intuitive concept of an algorithm being "better" than another one, if its proximity target curve is below that of the other one (that is, the target value for it is always smaller than the target value of the other one for the same proximity value). Such may not always be the case, the two proximity curves may cross each other, providing us with intervals of proximity values within which one or the other method is better.

While this way of thinking is not common in numerical analysis and in optimization, it is quite generally used in many sciences in situations where it is desirable to obtain an object for which the values of two evaluating functions are small simultaneously. A prime example is in estimation theory where we desire an estimation method with both small bias and small variance. More specifically, is the concept of a **receiver operating characteristics (ROC) curve** that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the true positive rate against the false positive rate at various threshold settings. One classifier system is considered "better" than the other one if its ROC curve is above that of the other one; but, just as for our proximity-target curves, the ROC curves for two classifier systems may cross each other. There are many publications on the role of ROC curves in the evaluation of medical imaging techniques; see, for example, [62, 74]. Their use for image reconstruction algorithm evaluation is discussed, for example, in [36].

For incarnations of the definitions the reader may wish to look ahead to Figure 1. That figure illustrates the discussed notions for two particular finite sequences $U := \left(x^k\right)_{k=K_{lo}}^{K_{hi}}$ and $V := \left(y^k\right)_{k=L_{lo}}^{L_{hi}}$. The details of how those sequences were specifically produced are given below in [21]. We use the notations of Subsection 4.2 and adapt the, more general, definitions of [21] to this case.

Definition 5.1. Monotone proximity of a finite sequence. Consider Θ such that $C_i \subseteq \Lambda \subseteq R^n$ for all i, and a proximity function $Prox_\Theta : \Lambda \to R_+$. For positive integers K_{lo} and $K_{hi} > K_{lo}$, the finite sequence $U := \left(x^k\right)_{k=K_{lo}}^{K_{hi}}$ of points in Λ is said to be **of monotone proximity** if for $K_{lo} < k \leq K_{hi}$, $Prox_\Theta\left(x^{k-1}\right) > Prox_\Theta\left(x^k\right)$.

Definition 5.2. The proximity-target curve of a finite sequence. Consider Θ such that $C_i \subseteq \Lambda \subseteq R^n$ for all i, a proximity function $Prox_\Theta : \Lambda \to R_+$, a target function $\phi : \Lambda \to R$, and positive integers K_{lo} and $K_{hi} > K_{lo}$. Let $U := \left(x^k\right)_{k=K_{lo}}^{K_{hi}}$ be a sequence of monotone proximity. Then the **proximity-target curve** $P \subseteq R^2$ associated with U is uniquely defined by:

(1) For $K_{lo} \le k \le K_{hi}$, $\left(Prox_\Theta\left(x^k\right), \phi\left(x^k\right)\right) \in P$.
(2) The intersection $\{(y,x) \in R^2 \mid Prox_\Theta\left(x^k\right) \le y \le Prox_\Theta\left(x^{k-1}\right)\} \cap P$ is the line segment from $\left(Prox_\Theta\left(x^{k-1}\right), \phi\left(x^{k-1}\right)\right)$ to $\left(Prox_\Theta\left(x^k\right), \phi\left(x^k\right)\right)$.

Definition 5.3. Comparison of proximity-target curves of sequences

Consider Θ such that $C_i \subseteq \Lambda \subseteq R^n$ for all i, a proximity function $Prox_\Theta : \Lambda \to R_+$, a target function $\phi : \Lambda \to R$, and positive integers K_{lo}, $K_{hi} > K_{lo}$, L_{lo}, $L_{hi} > L_{lo}$, let $R := \left(x^k\right)_{k=K_{lo}}^{K_{hi}}$ and $S := \left(y^k\right)_{k=L_{lo}}^{L_{hi}}$ be sequences of points in Ω of monotone proximity for which P and Q are their respective associated proximity-target curves. Define

$$\begin{aligned} t &:= \max\left(Prox_\Theta\left(x^{K_{hi}}\right), Prox_\Theta\left(y^{L_{hi}}\right)\right), \\ u &:= \min\left(Prox_\Theta\left(x^{K_{lo}}\right), Prox_\Theta\left(y^{L_{lo}}\right)\right). \end{aligned} \tag{5.1}$$

Then R **is better targeted than** S if:

(1) $t \le u$ and
(2) for any real number h, if $t \le h \le u$, $(h,v) \in P$ and $(h,w) \in Q$, then $v \le w$.

This definition is intuitively desirable. Suppose that we have an iterative algorithm that produces a sequence, $y^0, y^1, y^2, \cdots$, of which $S := \left(y^k\right)_{k=L_{lo}}^{L_{hi}}$ is a subsequence. An alternative algorithm that produces a sequence of points of which $R := \left(x^k\right)_{k=K_{lo}}^{K_{hi}}$ is a subsequence that is better targeted than S has a desirable property: Within the range $[t,u]$ of proximity values, the point that is produced by the alternative algorithm with that proximity value, is likely to have a lower (and definitely not higher) value of the target function than the point with that proximity value that is produced by the original algorithm. This property is stronger than what we stated before, namely that superiorization produces an output that is equally good from the point of view of proximity, but is superior with respect to the target function. Here the single output determined by a fixed ε is replaced by a set of potential outputs for any $\varepsilon \in [t,u]$.

6. The Superiorization Methodology Guarantee Problem

It is clear from the above text that the SM is interlacing the iterative steps of two separate and independent iterative processes. Therefore, we reformulate here **the guarantee problem of the SM** in the following general terms. One iterative process, the basic algorithm, is of the form

$$x^0 \in R^n,\ x^{k+1} = \mathscr{A}(x^k),\ k \ge 0, \tag{6.1}$$

where $\mathscr{A} : R^n \to R^n$ is a given algorithmic operator and $x^0 \in R^n$ is an arbitrary initialization point. It is assumed that this is a convergent process

$$\lim_{k\to\infty} x^k = x^* \in S \subseteq R^n, \tag{6.2}$$

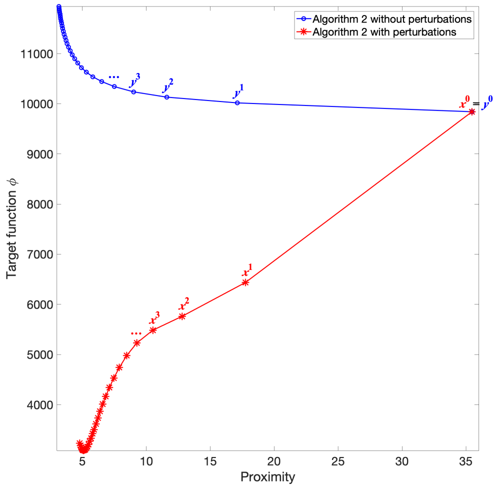

FIGURE 1. Proximity-target curves P and Q of the first 30 iterates of a feasibility-seeking algorithm (Algorithm 2 in [21]) with perturbations ($*$) and without perturbations ($\circ$). Reproduced from [21].

where $S \neq \varnothing$ and S is not a singleton. S could be a solution set of some problem, e.g., the convex feasibility problem. It is further assumed that the basic algorithm is bounded perturbations resilient as in Definition 4.1.

The second iterative process, henceforth called "the auxiliary algorithm", is

$$z^0 \in R^n,\ z^{k+1} = \mathscr{B}(z^k),\ k \geq 0, \tag{6.3}$$

where $\mathscr{B} : R^n \to R^n$ is another given algorithmic operator and $z^0 \in R^n$ is an arbitrary initialization point. It is assumed that this is also a convergent process

$$\lim_{k\to\infty} z^k = z^* \in T \subseteq R^n, \tag{6.4}$$

where $T \neq \varnothing$ and T is not a singleton. T could be a solution set of some other problem.

In these circumstances a general description of the SM consists of perturbing the iterates of the basic algorithm (6.1) by perturbations induced by the auxiliary algorithm (6.3), i.e.,

$$v^k := \frac{\mathscr{B}(y^k) - y^k}{\left\|\mathscr{B}(y^k) - y^k\right\|},\ \text{if } \left\|\mathscr{B}(y^k) - y^k\right\| \neq 0, \text{ and } v^k := 0, \text{ otherwise.} \tag{6.5}$$

This yields the superiorized version of the basic algorithm

$$y^{k+1} = \mathscr{A}\left(y^k + \beta_k v^k\right),\ \text{for all } k \geq 0, \tag{6.6}$$

with v^k as in (6.5) and $\{\beta_k\}_{k=0}^{\infty}$ such that $\beta_k \geq 0$, for all $k \geq 0$, and $\sum_{k=0}^{\infty} \beta_k < \infty$.

Example. If the operator $\mathscr{A}$ is a feasibility-seeking algorithm for a given convex feasibility problem and $\mathscr{B}$ is an unconstrained gradient descent algorithm for a given objective function $f : R^n \to R$ then (6.6) describes the earlier presented SM algorithms.

In [23, Section 3] we gave a precise definition of **the guarantee problem of the SM**. We wrote there: "The SM interlaces into a feasibility-seeking basic algorithm target function reduction steps. These steps cause the target function to reach lower values locally, prior to performing the next feasibility-seeking iterations. A mathematical guarantee has not been found to date that the overall process of the superiorized version of the basic algorithm will not only retain its feasibility-seeking nature but also accumulate and preserve globally the target function reductions."

Definition 6.1. The guarantee problem of the SM. Under which conditions one can guarantee that a superiorized version of a bounded perturbation resilient feasibility-seeking algorithm converges to a feasible point that has target function value smaller or equal to that of a point to which this algorithm would have converged if no perturbations were applied – everything else being equal.

Numerous works that are cited in [15] show that this global function reduction of the SM occurs in practice in many real-world applications. In addition to a partial answer in [23] with the aid of **the concentration of measure principle** there is also the partial result of [31, Theorem 4.1] about **strict Fejér monotonicity** of sequences generated by an SM algorithm.

Proving mathematically a guarantee of global function reduction of the SM will probably require some additional assumptions on the feasible set, on the objective function, on the parameters involved, or even on the set of permissible initialization points.

7. Some Applications of Superiorization

The, compiled and continuously updated, bibliography [15] contains references to numerous about superiorization (174 items as of August 17, 2023). We single out a few that applied the SM successfully to significant real-world problems. For many more theoretical, experimental studies or papers dealing with a variety of real-world applications, we recommend [15].

In [47], Guenter et al. considered the fully-discretized modeling of **image reconstruction from projections** problem that leads to a system of linear equations which is huge and very sparse. Solving such systems, sometimes under limitations on the computing resources, has been, is, and will remain a challenge. The authors aim not only at solving the linear system resulting from the modeling alone but consider the constrained optimization problem of minimizing an objective function subject to the modeling constraints. To do so, they recognized two fundamental approaches: (i) superiorization, and (ii) regularization. Within these two methodological approaches they evaluated 21 algorithms over a collection of 18 different phantoms (i.e., test problems), presenting their experimental results in very informative ways.

In [42], Fink et al. studied the **nonconvex multi-group multicast beamforming problem** with quality-of-service constraints and per-antenna power constraints. They formulated a convex relaxation of the problem as a semidefinite program in a real Hilbert space, which allowed them to approximate a point in the feasible set by iteratively applying a bounded perturbation resilient fixed-point mapping. Inspired by the superiorization methodology, they used this mapping as a basic algorithm, and added in each iteration a small perturbation with the intent to reduce the objective value and the distance to nonconvex rank-constraint sets.

Pakkaranang et al. [66] constructed a novel algorithm for solving **non-smooth composite optimization problems**. By using an inertial technique, they proposed a modified proximal gradient algorithm with outer perturbations and obtain strong convergence results for finding a solution of composite optimization problem. Based on bounded perturbation resilience, they presented their algorithm with the superiorization method and applied it to image recovery problem. They provided numerical experiments that demonstrated the efficiency of the algorithm and compared it with previously known algorithms in signal recovery.

In [6], the SM was applied to the **intensity-modulated radiation therapy (IMRT) treatment planning** problem. It was found there that the superiorization prototype solved the linearly constrained planning problem with similar performance to that of a general purpose nonlinear constrained optimizer while showing smooth convergence in both constraint proximity and objective function reduction. The authors' work demonstrated that superiorization is a useful alternative to constrained optimization in radiotherapy inverse treatment planning.

Especially interesting is the recent work of Ma et al. [60] who proposed a novel decomposition framework for **derivative-free optimization (DFO) algorithms** which significantly extended the scope of current DFO solvers to larger-scale problems. They proved that their proposed framework closely relates to the superiorization methodology that has been used for improving the efficiency of feasibility-seeking algorithms for constrained optimization problems in a derivative-based setting.

8. Concluding Remarks

In many mathematical formulations of significant real-world technological or physical problems, the objective function is exogenous to the modeling process which defines the constraints. In such cases, the faith of the modeler in the usefulness of an objective function for the application at hand is limited and, as a consequence, it is probably not worthwhile to invest too much resources in trying to reach an exact constrained minimum point. This is an argument in favor of using the superiorization methodology for practical applications. In doing so the amount of computational efforts invested alternatingly between performing perturbations and applying the basic algorithm's algorithmic operator can, and needs to, be carefully controlled in order to allow both activities to properly influence the outcome. Better theoretical insights into the behavior of weak and of strong superiorization as well as better ways of implementing the methodology are needed and await to be developed.

Acknowledgments

I am indebted to my partners in the work on suepriorization, past and present, from whom I have learned so much. In particular, my thanks go to the collaborators on the papers on superiorization that are referenced in this paper. Work on this paper was supported by the ISF-NSFC joint research program Grant No. 2874/19.

References

[1] M.S. Andersen and P.C. Hansen, Generalized row-action methods for tomographic imaging, Numerical Algorithms, 67 (2014), 121–144.

[2] F.J. Aragón-Artacho, Y. Censor, A. Gibali and D. Torregrosa-Belén, The superiorization method with restarted perturbations for split minimization problems with an application to radiotherapy treatment planning, Applied Mathematics and Computation, 440 (2023), 127627.

[3] C, Audet and J.E. Dennis Jr., A progressive barrier for derivative-free nonlinear programming, SIAM Journal on Optimization, 20 (2009), 445–472.

[4] C. Audet and W. Hare, Derivative-Free and Blackbox Optimization, Springer, Cham, 2017.

[5] C. Bargetz, S. Reich and R. Zalas, Convergence properties of dynamic string-averaging projection methods in the presence of perturbations, Numerical Algorithms, 77 (2018), 185–209.

[6] F. Barkmann, Y. Censor and N. Wahl, Superiorization as a novel strategy for linearly constrained inverse radiotherapy treatment planning, Preprint, 2022. Available on arXiv at: https://arxiv.org/abs/2207.13187.

[7] H.H. Bauschke and J.M. Borwein, On projection algorithms for solving convex feasibility problems, SIAM Review, 38 (1996), 367–426.

[8] H.H. Bauschke and P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Space, Springer, 2nd edition, 2017.

[9] H.H. Bauschke and V.R. Koch, Projection methods: Swiss army knives for solving feasibility and best approximation problems with halfspaces, Contemporary Mathematics, 636 (2015), 1–40.

[10] Bonacker, A. Gibali, K-H. Küfer and P. Süss, Speedup of lexicographic optimization by superiorization and its applications to cancer radiotherapy treatment, Inverse Problems, 33 (2017), 044012.

[11] D. Butnariu, R. Davidi, G.T. Herman, and I.G. Kazantsev, Stable convergence behavior under summable perturbations of a class of projection methods for convex feasibility and optimization problems, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), 540–547.

[12] D. Butnariu, S. Reich and A.J. Zaslavski, Convergence to fixed points of inexact orbits of Bregman-monotone and of nonexpansive operators in Banach spaces, in: H.F. Nathansky, B.G. de Buen, K. Goebel, W.A. Kirk, and B. Sims (Editors), Fixed Point Theory and its Applications, (Conference Proceedings, Guanajuato, Mexico, 2005), pp. 11–32, Yokahama Publishers, Yokahama, 2006.

[13] D. Butnariu, S. Reich and A.J. Zaslavski, Stable convergence theorems for infinite products and powers of nonexpansive mappings, Numerical Functional Analysis and Optimization 29 (2008), 304–323.

[14] A. Cegielski, Iterative Methods for Fixed Point Problems in Hilbert Spaces, Springer, Berlin, Heidelberg, Germany, 2012.

[15] Y. Censor, Superiorization and perturbation resilience of algorithms: A bibliography compiled and continuously updated, Available at: http://math.haifa.ac.il/yair/bib-superiorization-censor.html#top.

[16] Y. Censor and A. Cegielski, Projection methods: An annotated bibliography of books and reviews, Optimization, 64 (2015), 2343–2358.

[17] Y. Censor, W. Chen, P.L. Combettes, R. Davidi and G.T. Herman, On the effectiveness of projection methods for convex feasibility problems with linear inequality constraints, Computational Optimization and Applications, 51 (2012), 1065–1088.

[18] Y. Censor, R. Davidi and G.T. Herman, Perturbation resilience and superiorization of iterative algorithms, Inverse Problems, 26 (2010), 065008.

[19] Y. Censor, T. Elfving and G.T. Herman, Averaging strings of sequential iterations for convex feasibility problems, in: D. Butnariu, Y. Censor and S. Reich (Editors), Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, pp. 101–114, Elsevier Science Publishers, Amsterdam, 2001.

[20] Y. Censor, R. Davidi, G.T. Herman, R.W. Schulte and L. Tetruashvili, Projected subgradient minimization versus superiorization, Journal of Optimization Theory and Applications, 160 (2014), 730–747.

[21] Y. Censor, E. Garduño, E.S. Helou and G.T. Herman, Derivative-free superiorization: principle and algorithm, Numerical Algorithms, 88 (2021), 227–248.

[22] Y. Censor, G.T. Herman and M. Jiang (Guest Editors), Superiorization: Theory and Applications, Inverse Problems, Vol. 33, No. 4, 2017.

[23] Y. Censor and E. Levy, An analysis of the superiorization method via the principle of concentration of measure, Applied Mathematics and Optimization, 83 (2021), 2273–2301.

[24] Y. Censor, H. Heaton and R.W. Schulte, Derivative-free superiorization with component-wise perturbations, Numerical Algorithms,80 (2019), 1219–1240.

[25] Y. Censor and A. Segal, On the string averaging method for sparse common fixed point problems, International Transactions in Operational Research, 16 (2009), 481–494.

[26] Y. Censor and A. Segal, On string-averaging for sparse problems and on the split common fixed point problem, Contemporary Mathematics, 513 (2010), 125–142.

[27] Y. Censor and E. Tom, Convergence of string-averaging projection schemes for inconsistent convex feasibility problems, Optimization Methods and Software, 18 (2003), 543–554.

[28] Y. Censor and M. Zaknoon, Algorithms and convergence results of projection methods for inconsistent feasibility problems: A review. Pure and Applied Functional Analysis, 3 (2018), 565–586.

[29] Y. Censor and A.J. Zaslavski, Convergence and perturbation resilience of dynamic string-averaging projection methods, Computational Optimization and Applications, 54 (2013), 65–76.

[30] Y. Censor and A.J. Zaslavski, String-averaging projected subgradient methods for constrained minimization, Optimization Methods & Software, 29 (2014), 658–670.

[31] Y. Censor and A.J. Zaslavski, Strict Fejér monotonicity by superiorization of feasibility-seeking projection methods, Journal of Optimization Theory and Applications, 165 (2015), 172–187.

[32] D.E. Ciaurri, O. Isebor and L. Durlofsky, Application of derivative-free methodologies to generally constrained oil production optimization problems, Procedia Computer Science, 1 (2012), 1301–1310.

[33] P.L. Combettes, Strong convergence of block-iterative outer approximation methods for convex optimization, SIAM Journal on Control and Optimization, 38 (2000), 538–565.

[34] P.L. Combettes, On the numerical robustness of the parallel projection method in signal synthesis, IEEE Signal Processing Letters, 8 (2001), 45–47.

[35] A.R. Conn, K. Scheinberg, and L.N. Vicente, Introduction to Derivative-Free Optimization, SIAM, Philadelphia, USA, 2009.

[36] T.A. Cooley and H.H. Barrett, Evaluation of statistical methods for image reconstruction through ROC analysis, IEEE Transactions on Medical Imaging, 11 (1992), 276–282.

[37] G. Crombez, Finding common fixed points of strict paracontractions by averaging strings of sequential iterations, Journal of Nonlinear and Convex Analysis, 3 (2002), 345–351.

[38] R. Davidi, Algorithms for Superiorization and their Applications to Image Reconstruction, Ph.D. dissertation, Department of Computer Science, The City University of New York, NY, USA, 2010.

[39] R. Davidi, G.T. Herman, and Y. Censor, Perturbation-resilient block-iterative projection methods with application to image reconstruction from projections, International Transactions in Operational Research, 16 (2009), 505–524.

[40] R. Davidi, Y. Censor, R.W. Schulte, S. Geneser and L. Xing, Feasibility-seeking and superiorization algorithms applied to inverse treatment planning in radiation therapy, Contemporary Mathematics, 636 (2015), 83–92.

[41] M. Diniz-Ehrhardt, J. Martínez, and L. Pedroso, Derivative-free methods for nonlinear programming with general lower-level constraints, Computational and Applied Mathematics, 30 (2011), 19–52.

[42] J. Fink, R.L.G. Cavalcante and S. Stanczak, Multi-group multicast beamforming by superiorized projections onto convex sets, IEEE Transactions on Signal Processing, 69 (2021), 5708–5722.

[43] E. Garduño, and G.T. Herman, Superiorization of the ML-EM algorithm, IEEE Transactions on Nuclear Science, 61 (2014), 162–172.

[44] H.A. Gay and A. Niemierko, A free program for calculating EUD-based NTCP and TCP in external beam radiotherapy, Physica Medica, 23 (2007), 115–125.

[45] A. Gibali, G.T. Herman and C. Schnörr, (Guest Editors), Superiorization versus Constrained Optimization: Analysis and Applications, Journal of Applied and Numerical Optimization, Vol. 2, No. 1, 2020.

[46] D. Gordon and R. Gordon, Component-averaged row projections: A robust, block-parallel scheme for sparse linear systems, SIAM Journal on Scientific Computing, 27 (2005), 1092–1117.

[47] M. Guenter, S. Collins, A. Ogilvy, W. Hare and A. Jirasek, Superiorization versus regularization: A comparison of algorithms for solving image reconstruction problems with applications in computed tomography, Medical Physics, 49 (2022), 1065–1082.

[48] H. He and H-K. Xu, Perturbation resilience and superiorization methodology of averaged mappings, Inverse Problems, 33 (2017), 044007.

[49] G.T. Herman, Fundamentals of Computerized Tomography: Image Reconstruction from Projections, 2nd Edition, Springer-Verlag, London, 2009.

[50] G.T. Herman, Superiorization for image analysis, in: Combinatorial Image Analysis, Lecture Notes in Computer Science, Vol. 8466, pp. 1-7, Springer, 2014.

[51] G.T. Herman, Problem structures in the theory and practice of superiorization, Journal of Applied and Numerical Optimization, 2 (2020), 71–76.

[52] G.T. Herman and R. Davidi, Image reconstruction from a small number of projections, Inverse Problems, 24 (2008), 045011.

[53] G.T. Herman, E. Garduño, R. Davidi and Y. Censor, Superiorization: An optimization heuristic for medical physics, Medical Physics, 39 (2012), 5532–5546.

[54] M. Hoseini, S. Saeidi and D.S. Kim, On perturbed hybrid steepest descent method with minimization or superiorization for subdifferentiable functions, Numerical Algorithms, 85 (2020), 353–374.

[55] W. Jin, Y. Censor and M. Jiang, A heuristic superiorization-like approach to bioluminescence, International Federation for Medical and Biological Engineering (IFMBE) Proceedings, 39 (2013), 1026–1029.

[56] J. Klukowska, R. Davidi, G.T. Herman, SNARK09–A software package for reconstruction of 2D images from 1D projections, Computer Methods and Programs in Biomedicine, 110 (2013), 424–440.

[57] O. Langthaler, Incorporation of the Superiorization Methodology into Biomedical Imaging Software, Marshall Plan Scholarship Report, Salzburg University of Applied Sciences, Salzburg, Austria, and The Graduate Center of the City University of New York, NY, USA, 2014.

[58] L. Li, Y. Chen, Q. Liu, J. Lazic, W. Luo, and Y. Li, Benchmarking and evaluating MATLAB derivative-free optimisers for single-objective applications, in: D.S. Huang, K.H. Jo, and J. Figueroa-García, (Editors), Intelligent Computing Theories and Application, ICIC 2017, pp. 75–88 Springer, 2017.

[59] S. Luo and T. Zhou, Superiorization of EM algorithm and its application in single-photon emission computed tomography (SPECT), Inverse Problems and Imaging, 8 (2014), 223–246.

[60] K. Ma, N.V. Sahinidis, S. Rajagopalan, S. Amaran and S.J. Bury, Decomposition in derivative-free optimization, Journal of Global Optimization, 81 (2021), 269–292.

[61] V. Martín-Márquez, S. Reich and S. Sabach, Iterative methods for approximating fixed points of Bregman nonexpansive operators, Discrete and Continuous Dynamical Systems - Series S, 6 (2012), 1043–1063.

[62] C.E. Metz, Some practical issues of experimental design and data analysis in radiological ROC studies, Investigative Radiology, 24 (1989), 234–243.

[63] T. Nikazad, M. Abbasi, L. Afzalipour and T. Elfving, A new step size rule for the superiorization method and its application in computerized tomography, Numerical Algorithms, 90 (2022), 1253–1277.

[64] T. Nikazad, R. Davidi and G.T. Herman, Accelerated perturbation-resilient block-iterative projection methods with application to image reconstruction, Inverse Problems, 28 (2012), 035005.

[65] H. Nystrom, M.F. Jensen and P.W. Nystrom, Treatment planning for proton therapy: what is needed in the next 10 years? The British Journal of Radiology, 93 (2020), 20190304.

[66] N. Pakkaranang, P. Kumam, V. Berinde and Y.I. Suleiman, Superiorization methodology and perturbation resilience of inertial proximal gradient algorithm with application to signal recovery, The Journal of Supercomputing, 76 (2020), 9456–9477.

[67] S.N. Penfold, R.W. Schulte, Y. Censor, V. Bashkirov, S. McAllister, K.E. Schubert and A.B. Rosenfeld, Block-iterative and string-averaging projection algorithms in proton computed tomography image reconstruction, in: Y. Censor, M. Jiang and G. Wang (Editors), Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems, pp. 347–367, Medical Physics Publishing, Madison, WI, 2010,

[68] S.N. Penfold, R.W. Schulte, Y. Censor and A.B. Rosenfeld, Total variation superiorization schemes in proton computed tomography image reconstruction, Medical Physics, 37 (2010), 5887–5895.

[69] B. Prommegger, Verification and Evaluation of Superiorized Algorithms Used in Biomedical Imaging: Comparison of Iterative Algorithms With and Without Superiorization for Image Reconstruction from Projections, Marshall Plan Scholarship Report, Salzburg University of Applied Sciences, Salzburg, Austria, and The Graduate Center of the City University of New York, NY, USA, 2014.

[70] H. Rhee, An application of the string averaging method to one-sided best simultaneous approximation, Journal of the Korean Society of Mathematical Education, Series B, 10 (2003), 49–56.

[71] L.M. Rios and N.V. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, Journal of Global Optimization, 56 (2013), 1247–1293.

[72] H.A. Simon, Rational choice and the structure of the environment, Psychological Review, 63 (1956), 129–138.

[73] SNARK14, A programming system for the reconstruction of 2D images from 1D projections. Released: 2015. Available at: https://turing.iimas.unam.mx/SNARK14M/index.php.

[74] J.A. Swets, ROC analysis applied to the evaluation of medical imaging techniques, Investigative Radiology, 14 (1979), 109–112.

[75] Q. Yang, W. Cong and G. Wang, Superiorization-based multi-energy CT image reconstruction, Inverse Problems, 33 (2017), 044014.

[76] A.J. Zaslavski, Algorithms for Solving Common Fixed Point Problems, Springer International Publishing AG, part of Springer Nature, 2018.