An. Şt. Univ. Ovidius Constanţa Vol. xx(x), 201x, 0–13

Weak and Strong Superiorization: Between Feasibility-Seeking and Minimization

Yair Censor

Abstract

We review the superiorization methodology, which can be thought of, in some cases, as lying between feasibility-seeking and constrained minimization. It is not quite trying to solve the full fledged constrained minimization problem; rather, the task is to find a feasible point which is superior (with respect to an objective function value) to one returned by a feasibility-seeking only algorithm. We distinguish between two research directions in the superiorization methodology that nourish from the same general principle: Weak superiorization and strong superiorization and clarify their nature.

1 Introduction

What is superiorization. The superiorization methodology works by taking an iterative algorithm, investigating its perturbation resilience, and then, using proactively such permitted perturbations, forcing 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 original algorithm* is computationally efficient and useful in terms of the application at hand, and if the perturbations are simple and not expensive

Key Words: perturbation resilience, constrained minimization, convex feasibility problem, dynamic string-averaging, superiorization methodology, superiorized version of an algorithm, strict Fejér monotonicity.

2010 Mathematics Subject Classification: 90C25, 90C30, 90C45, 65K10.

Received: November 27, 2014.

Revised: February 5, 2015.

Accepted: xxxxxx.

*We use the term "algorithm" for the iterative processes discussed here, even for those that do not include any termination criterion. 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.

to calculate, then the advantage of this methodology is that, for essentially the computational cost of the original 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 some important practical applications and awaits to be implemented and tested in additional fields; see, e.g., the recent papers [21, 34], for applications in intensity-modulated radiation therapy and in nondestructive testing. Although not limited to this case, an important special case of the superiorization methodology is when the original algorithm is a feasibility-seeking algorithm, or one that strives to find constraint-compatible points for a family of constraints, and the perturbations that are interlaced into the original algorithm aim at reducing (not necessarily minimizing) a given merit (objective) function. We distinguish between 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 (nonempty intersection) 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[†].

The purpose of this paper. 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. A recent review paper on the subject which should be read together with this paper is Herman's [25]. Unless otherwise stated, we restrict ourselves, for simplicity, to the J-dimensional Euclidean space R^J although some materials below remain valid in Hilbert space.

Superiorization related work. Recent publications on the superiorization methodology (SM) are devoted to either weak or strong superiorization, without yet using these terms. They are [2, 3, 8, 14, 20, 21, 22, 26, 27, 28, 30,

[†]These terms were proposed in [16], following a private discussion with our colleague and coworker in this field Gabor Herman.

32], culminating in [34] and [10]. The latter contains a detailed description of the SM, its motivation, and an up-to-date review of SM-related previous works scattered in earlier publications, including a reference to [3] in which it all started, although without using yet the terms superiorization and perturbation resilience. [3] was the first to propose this approach and implement it in practice, but its roots go back to [4, 5] 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, see also [19]. Bounded perturbation resilience of a parallel projection method (PPM) was observed as early as 2001 in [17, Theorem 2] (without using this term). More details on related work appear in [10, Section 3] and in [15, Section 1].

2 The framework

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

Case 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 $\Psi_T = \{x^* \in \Phi \mid f(x^*) \leq f(x)$ for all $x \in \Phi\}$.*

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

Problem 3. ***The Function Reduction Problem****. Let $\Psi_T \subseteq R^J$ be the solution set of some given mathematically-formulated problem T and let $\phi : R^J \to R$ be an objective function. Let $\mathcal{A} : R^J \to R^J$ 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 lesser 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. The so obtained vector x^* need not be a minimizer of ϕ over Ψ_T. Another point to observe is that the very problem formulation itself 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 $\mathcal{A}$, for the solution of problem T, and its superiorized version.

A fundamental difference between weak and strong superiorization lies in the meaning attached to the term "solution of problem T" in Problem 3. 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 1. Superiorization work on Case 2, where T is a maximum likelihood optimization problem and Ψ_T – its solution set, appears in [22, 28, 29]. To present the principles of weak superiorization (in Section 3 below) we use the Dynamic String-Averaging Projection (DSAP) method of [14] while the strong superiorization is demonstrated on a general basic algorithm in Section 4.

3 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. Convergence questions are studied in their asymptotically. 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. ***Bounded perturbation resilience (BPR).*** *Let $\Gamma \subseteq R^J$ be a given nonempty set. An algorithmic operator $\mathcal{A} : R^J \rightarrow R^J$ 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 $x^{k+1} = \mathcal{A}(x^k)$, for all $k \geq 0$, converges to a point in Γ for all $x^0 \in R^J$, then any sequence $\{y^k\}_{k=0}^{\infty}$ of points in R^J that is generated by $y^{k+1} = \mathcal{A}(y^k + \beta_k v^k)$, for all $k \geq 0$, also converges to a point in Γ for all $y^0 \in R^J$ 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 $\phi : R^J \rightarrow R$ 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 $\Gamma = R^J$. In other specific cases care must be taken regarding how Γ and Ψ_T are related. The following Superiorized Version of the Basic Algorithm $\mathcal{A}$ is based on [16, Algorithm 4.1].

Algorithm 5. ***Superiorized Version of the Basic Algorithm*** $\mathcal{A}$.

(0) Initialization*: Let N be a natural number and let $y^0 \in R^J$ be an arbitrary user-chosen vector.*

(1) 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:*

(1.1) Inner loop initialization*: Define $y^{k,0} = y^k$.*

(1.2) Inner loop step: *Given $y^{k,n}$, as long as $n < N_k$, do as follows:*

(1.2.1) *Pick a $0 < \beta_{k,n} \leq 1$ in a way that guarantees that*

$$\sum_{k=0}^{\infty} \sum_{n=0}^{N_k - 1} \beta_{k,n} < \infty. \tag{1}$$

(1.2.2) *Pick an $s^{k,n} \in \partial\phi(y^{k,n})$ and define $v^{k,n}$ as follows:*

$$v^{k,n} = \begin{cases} -\dfrac{s^{k,n}}{\left\| s^{k,n} \right\|}, & \text{if } 0 \notin \partial\phi(y^{k,n}), \\ 0, & \text{if } 0 \in \partial\phi(y^{k,n}). \end{cases} \tag{2}$$

(1.2.3) *Calculate the perturbed iterate*

$$y^{k,n+1} = y^{k,n} + \beta_{k,n} v^{k,n} \tag{3}$$

and if $n + 1 < N_k$ set $n \leftarrow n + 1$ and go to ***(1.2)****, otherwise go to* ***(1.3)****.*

(1.3) *Exit the inner loop with the vector y^{k,N_k}*

(1.4) *Calculate*

$$y^{k+1} = \mathcal{A}(y^{k,N_k}) \tag{4}$$

set $k \leftarrow k + 1$ and go back to ***(1)****.*

Let us consider Case 1 in Section 2 wherein T is a convex feasibility problem. The Dynamic String-Averaging Projection (DSAP) method of [14] constitutes a family of algorithmic operators that can play the role of the above $\mathcal{A}$ in a Basic Algorithm for the solution of the CFP T.

Let $C_1, C_2, \ldots, C_m$ be nonempty closed convex subsets of a Hilbert space X where m is a natural number. 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 $t_i \in \{1, 2, \ldots, m\}$ for all $i = 1, 2, \ldots, q$, whose length is $\ell(t) = q$. The product of the individual projections onto the sets whose indices appear in the index vector t is $P[t] := P_{t_q} \cdots P_{t_1}$, 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 $\mathcal{M}$ 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 $(\Omega, w) \in \mathcal{M}$ define the convex combination of the end-points of all strings defined by members of Ω

$$P_{\Omega,w}(x) := \sum_{t \in \Omega} w(t) P[t](x),\ x \in X. \tag{5}$$

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

$$\mathcal{M}_* = \{(\Omega, w) \in \mathcal{M} \mid \ell(t) \leq \bar{q} \text{ and } w(t) \geq \Delta,\ \forall\ t \in \Omega\}. \tag{6}$$

Algorithm 6. ***The DSAP method with variable strings and variable weights Initialization****: select an arbitrary $x^0 \in X$,* ***Iterative step****: given a current iteration vector x^k pick a pair $(\Omega_k, w_k) \in \mathcal{M}_*$ and calculate the next iteration vector x^{k+1} by*

$$x^{k+1} = P_{\Omega_k, w_k}(x^k). \tag{7}$$

The first prototypical string-averaging algorithmic scheme appeared in [9] and subsequent work on its realization with various algorithmic operators includes [11, 12, 13, 15, 18, 23, 31, 32, 33]. 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 (for Algebraic Reconstruction Technique) in image reconstruction from projections, see [24]. If, at the other extreme, one uses exactly m one-dimensional 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 [1, 6, 7] for more information on projection methods.

The superiorized version of the DSAP algorithm is obtained by using Algorithm 6 as the algorithmic operator $\mathcal{A}$ in Algorithm 5. 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 7. *[16, 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 $||x - y|| \leq r_0$,*

$$|\phi(x) - \phi(y)| \leq \bar{L}||x - y||, \tag{8}$$

and suppose that $\{(\Omega_k, w_k)\}_{k=0}^{\infty} \subset \mathcal{M}_*$. *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 each integer* $k \geq k_0$,

$$\|y^{k+1} - x\|^2 \leq \|y^k - x\|^2 - c_0 \sum_{n=1}^{N_k - 1} \beta_{k,n}. \tag{9}$$

This shows 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 it 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 reduction of the value of the objective function ϕ is indeed achieved, without loosing the convergence toward feasibility, see [2, 3, 8, 14, 20, 21, 22, 26, 27, 28, 30, 32]. 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., [10, Table 1].

4 Strong superiorization

As in the previous section, let us consider again, Case 1 in Section 2 wherein T is a convex feasibility problem. In this section we present a restricted version of the SM of [27] as adapted to this situation in [10]. Let $C_1, C_2, \ldots, C_m$ be nonempty closed convex subsets of R^J where m is a natural number 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 $\Lambda \subset R^J$ such that $C_i \subset \Lambda$ for all $i = 1, 2, \ldots, m$, thus $C = \cap_{i=1}^I C_i \subset \Lambda$. Instead of the nonemptiness assumption we associate with the family of constraints $\{C_i\}_{i=1}^m$ a *proximity function* $Prox_C : \Lambda \to \mathbb{R}_+$ 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 $Prox_C(x) \leq \varepsilon$ is called an *ε-compatible solution* for C. We further assume that we have a feasibility-seeking *algorithmic operator* $\mathcal{A} : R^J \to \Lambda$, with which we define the Basic Algorithm as the iterative process

$$x^{k+1} = \mathcal{A}(x^k), \text{ for all } k \geq 0, \text{ for an arbitrary } x^0 \in \Lambda. \tag{10}$$

The following definition helps to evaluate the output of the Basic Algorithm upon termination by a stopping rule. This definition as well as most of the remainder of this section appeared in [27].

Definition 8. ***The ε-output of a sequence.*** *Given $C \subseteq \Lambda \subseteq R^J$, a proximity function $Prox_C : \Lambda \rightarrow R_+$, 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) $Prox_C\left(x^K\right) \leq \varepsilon$, and (ii) $Prox_C\left(x^k\right) > \varepsilon$ 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** *$(C,\ Prox_C)$.*

We denote the ε-output by $O\left(C,\varepsilon,\left\{x^k\right\}_{k=0}^{\infty}\right) = x^K$. Clearly, an ε-output $O\left(C,\varepsilon,\left\{x^k\right\}_{k=0}^{\infty}\right)$ 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 $O\left(C,\varepsilon,\left\{x^k\right\}_{k=0}^{\infty}\right)$ is the *output* produced by that algorithm when it includes the termination rule to stop when an ε-compatible solution for C is reached.

Definition 9. ***Strong perturbation resilience.*** *Assume that we are given a $C \subseteq \Lambda$, a proximity function $Prox_C$, an algorithmic operator $\mathcal{A}$ 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 $O\left(C,\varepsilon,\left\{x^k\right\}_{k=0}^{\infty}\right)$ exists for every $x^0 \in \Lambda$; (ii) for every $\varepsilon > 0$, for which the ε-output $O\left(C,\varepsilon,\left\{x^k\right\}_{k=0}^{\infty}\right)$ exists for every $x^0 \in \Lambda$, we have also that the ε'-output $O\left(C,\varepsilon',\left\{y^k\right\}_{k=0}^{\infty}\right)$ exists for every $\varepsilon' > \varepsilon$ and for every sequence $\left\{y^k\right\}_{k=0}^{\infty}$ generated by*

$$y^{k+1} = \mathcal{A}\left(y^k + \beta_k v^k\right),\ \textit{for all}\ k \geq 0, \tag{11}$$

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 [27, Theorem 1]. Along with the $C \subseteq R^J$, we look at the objective function $\phi : R^J \rightarrow R$, with the convention that a point in R^J for which the value of ϕ is smaller is considered *superior* to a point in R^J for which the value of ϕ is larger. The essential idea of the SM is to make use of the perturbations of (11) to transform a strongly perturbation resilient Basic Algorithm that seeks a constraints-compatible solution for C 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 10. *Given a function $\phi : R^J \to R$ and a point $y \in R^J$, we say that a vector $d \in R^J$ 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* $\phi(y + \lambda d) \leq \phi(y)$.

Obviously, the zero vector is always such a vector, but for superiorization to work we need a sharp inequality to occur in (10) 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 Superiorized Version of the Basic Algorithm is presented next by its pseudo-code.

Algorithm 11. ***Superiorized Version of the Basic Algorithm***

1. **set** $k = 0$
2. **set** $y^k = y^0$
3. **set** $\ell = -1$
4. **repeat**
5.   **set** $n = 0$
6.   **set** $y^{k,n} = y^k$
7.   **while** $n{<}N$
8.   **set** $v^{k,n}$ to be a nonascending vector for ϕ at $y^{k,n}$
9.   **set** *loop=true*
10.   **while** *loop*
11.   **set** $\ell = \ell + 1$
12.   **set** $\beta_{k,n} = \eta_\ell$
13.   **set** $z = y^{k,n} + \beta_{k,n} v^{k,n}$
14.   **if** $\phi(z) {\leq} \phi\left(y^k\right)$ **then**
15.   **set** $n{=}n + 1$

16. **set** $y^{k,n}=z$

17. **set** $loop = false$

18. **set** $y^{k+1}=\mathcal{A}\left(y^{k,N}\right)$

19. **set** $k = k + 1$

Theorem 12. *Any sequence $\left\{y^k\right\}_{k=0}^{\infty}$, generated by (the Superiorized Version of the Basic Algorithm) Algorithm 11, satisfies (11). Further, if, for a given $\varepsilon > 0$, the ε-output $O\left(C, \varepsilon, \left\{x^k\right\}_{k=0}^{\infty}\right)$ 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 11, has an ε'-output $O\left(C, \varepsilon', \left\{y^k\right\}_{k=0}^{\infty}\right)$ for every $\varepsilon' > \varepsilon$.*

Proof. The proof of this theorem follows from the analysis of the behavior of the Superiorized Version of the Basic Algorithm in [27, pp. 5537–5538] and is, therefore, not repeated here. ■

In other words, the theorem says that Algorithm 11 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 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.

Algorithms 5 and 11 are not identical. For example, the first employes negative subgradients while the second allows to use any nonascending directions of ϕ. Nevertheless, they are based on the same leading principle of the superiorization methodology. Comments on the differences between them can be found in [16, Remark 4.1]. While experimental work has repeatedly demonstrated benefits of the SM, the Theorems 7 and 12 related to these superiorized versions of the Basic Algorithm, respectively, leave much to be desired in terms of rigorously analyzing the behavior of the SM under various conditions.

5 Concluding comments

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.

Additional questions that come to mind but have not yet been addressed so far are related to possible extensions of the superiorization methodology such as: non-convex objective function ϕ, other control sequences (beyond cyclic and almost cyclic) and possibly random control sequences; infinitely many sets C_i; Hilbert space formulations and more.

Acknowledgments. This paper was presented at the Tenth Workshop on Mathematical Modelling of Environmental and Life Sciences Problems, October 16-19, 2014, Constantza, Romania. We thank two anonymous referees for their insightful comments. The work was supported by Research Grant No. 2013003 of the United States-Israel Binational Science Foundation (BSF).

References

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

[2] H.H. Bauschke and V.R. Koch, Projection methods: Swiss army knives for solving feasibility and best approximation problems with half-spaces, *Contemporary Mathematics*, accepted for publication. https://people.ok.ubc.ca/bauschke/Research/c16.pdf.

[3] 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.

[4] 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, *Fixed Point Theory and its Applications*, (Conference Proceedings, Guanajuato, Mexico, 2005), Yokahama Publishers, Yokahama, Japan, pp. 11–32, 2006.

[5] 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.

[6] Y. Censor and A. Cegielski, Projection methods: an annotated bibliography of books and reviews, *Optimization*, accepted for publication. DOI:10.1080/02331934.2014.957701.

[7] 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.

[8] Y. Censor, R. Davidi and G.T. Herman, Perturbation resilience and superiorization of iterative algorithms, *Inverse Problems* **26** (2010), 065008 (12pp).

[9] 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*, Elsevier Science Publishers, Amsterdam, 2001, pp. 101–114.

[10] 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.

[11] 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.

[12] 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.

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

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

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

[16] Y. Censor and A.J. Zaslavski, Strict Fejér monotonicity by superiorization of feasibility-seeking projection methods, *Journal of Optimization*

Theory and Applications, accepted for publication. DOI:10.1007/s10957-014-0591-x.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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*, accepted for publication. http://math.haifa.ac.il/yair/con-math-DCSGX-final-300114.pdf

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

[23] 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.

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

[25] G.T. Herman, Superiorization for image analysis, in: *Combinatorial Image Analysis*, Lecture Notes in Computer Science Vol. **8466**, Springer, 2014, pp. 1–7. DOI: 10.1007/978-3-319-07148-0_1.

[26] G.T. Herman and R. Davidi, Image reconstruction from a small number of projections, *Inverse Problems* **24** (2008), 045011 (17pp).

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

[28] 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.

[29] 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.

[30] 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 (19pp).

[31] 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*, Medical Physics Publishing, Madison, WI, USA, 2010, pp. 347–367.

[32] 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.

[33] 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, Pure and Applied Mathematics* **10** (2003), 49–56.

[34] M.J. Schrapp and G.T. Herman, Data fusion in X-ray computed tomography using a superiorization approach, *Review of Scientific Instruments* **85** (2014), 053701 (9pp).

Yair CENSOR,
Department of Mathematics,
University of Haifa,
Mt. Carmel, Haifa 3498838, Israel.
Email: yair@math.haifa.ac.il