Forcing optimality and Brandt's principle

Domenico Napoletani, Marco Panza, Daniele Struppa

Forcing optimality and Brandt's principle

Domenico Napoletani, Marco Panza and Daniele C. Struppa

Abstract We argue that many optimization methods can be viewed as representatives of "forcing", a methodological approach that attempts to bridge the gap between data and mathematics on the basis of an *a priori* trust in the power of a mathematical technique, even when detailed, credible models of a phenomenon are lacking or do not justify the use of this technique. In particular, we show that forcing is implied in particle swarms optimization methods, and in modeling image processing problems through optimization. From these considerations, we extrapolate a principle for general data analysis methods, what we call 'Brandt's principle', namely the assumption that an algorithm that approaches a steady state in its output has found a solution to a problem, or needs to be replaced. We finally propose that biological systems, and other phenomena that respect general rules of morphogenesis, are a natural setting for the application of this principle.

1 Introduction

In a series of previous papers [26, 27, 28, 29] we described what we called the 'microarray paradigm' and we showed that there are specific methodological motifs that structure the approach of data analysis to scientific problems. By 'microarray paradigm' we referred to the belief that sufficiently large data collected from a phe-

Domenico Napoletani
Institute for Quantum Studies, Chapman University, Orange, CA, 92866, e-mail: napoleta@chapman.edu

Marco Panza
CNRS, IHPST (UMR 8590 of CNRS, University of Paris 1 Panthéon-Sorbonne), and Chapman University, Orange, CA, 92866, e-mail: marco.panza@univ-paris1-fr

Daniele C. Struppa
Schmid College of Science and Technology, Chapman University, Orange, CA 92866, e-mail: struppa@chapman.edu

nomenon allows answering any question about the phenomenon itself. Answers are then found through a process of automatic fitting of the data to models that do not carry any structural understanding beyond the actual solution of the problem. This is a practice we suggested to label 'agnostic science'.

We argued as well in [26] that, in data analysis, mathematics is "forced" onto the data. By this we mean that there are techniques expected to be useful, even when the assumptions under which these techniques should be applied do not appear to hold for the phenomenon under study. This process, which we called 'forcing', can be viewed as a direct, coarse, and willful attempt to bridge the gap between data and mathematics. This agnostic approach displays a role of mathematics in science that is essentially different from the traditional one: rather than offering a structured interface between our problems and the raw data, mathematics now provides a rich repository of techniques for forcing. The extent to which forcing is possible, and therefore the relation between specific classes of phenomena and specific mathematical techniques to force onto them, suggests a shift in the focus of our understanding: from the phenomenon itself and its inner structure, to the structure of the algorithmic processes that are privileged in data analysis. The consequence is that the link between reasonable heuristic assumptions and successful problem solving through data analysis can be broken without impact on the effectiveness of the problem solving itself.

Here, we will show the implications of this shift by dissecting one of its most pervasive aspects: the search for optimization and the belief that optimization as such is always useful and necessary. For our purposes, to solve an optimization problem means to find the minimum of a fitness function $F(x)$ on a given domain $\mathscr{S}$ (usually a subset of $\mathbb{R}^n$ for some $n > 0$). We explore how forcing appears in the ways the fitness function $F(x)$ is built starting from a specific empirical problem, and in the methods used to approach the optimization problem itself.

We also propose that the way optimization techniques are used in practice often hints at a more basic methodological principle in data analysis, what we call 'Brandt's Principle', which can be expected to have wide applicability in problems from life and social sciences. The articulation of this suggestion goes through three sections. In Sect. 2 we explore how optimization can be seen as forcing. In Sect. 3, we draw the extreme consequence of regarding optimization as forcing; we suggest that, quite often, the local behavior of optimization algorithms is more important than its ultimate convergence to any optimal solution to a problem, and we introduce Brandt's principle, an operative principle of this approach. In Sect. 4, we speculate on the reasons of the effectiveness of this principle, particularly when applied to solve problems about what we suggest to call 'historical phenomena', i.e. phenomena significantly constrained by their past development. To this end, we first analyze a recently proposed principle of morphogenesis in developmental biology [25], and then explore its strong analogy with Brandt's principle.

2 Forcing optimality

When using optimization to solve an empirical problem, there are two levels at which we may have forcing.

At one level, it is possible that the optimization method is forced upon the empirical problem: though the fitness function may or may not encode significant information on the problem and the underlying phenomenon, the method used to solve the optimization problem does not actually find the global extrema of the function, but nevertheless the original empirical problem is solved by the (non-optimal) output of the algorithm. To show an example of this situation, we analyze in Sect. 2.1 a popular method of optimization in data analysis, particle swarms optimization (PSO). Specifically, we show that its effectiveness is not bound to the actual ability of this algorithm to find a globally optimal solution to a problem, rather, PSO can be seen as a collection of local explorations of the solution space.

At another level, the fitness function is forced upon the empirical problem, even when poorly related to it. The more complex the empirical problem, the more the associated fitness function will be *ad hoc*, and many fitness functions, not all leading to the same optimum solutions, are possible. And yet, despite the *ad hoc* nature of the fitness function, the empirical problem is successfully solved. To explore this second level of forcing, we show in Sect. 2.2 how important problems, such as image processing, lead to entire families of fitness functions with different choices of parameters.

In Sect. 2.2 we also show that the complexity of these problems and associated fitness functions is such that the whole variety of optimization methods used to solve the corresponding optimization problems often reduce to a local, point by point search for solutions, with no hope in general for the identification of the global optimum, similarly to what we argued in Sect. 2.1 for the restricted case of PSO methods. We can imply therefore that forcing the fitness function usually leads to forcing the optimization method itself.

We conclude that, in data analysis, what is generally relevant in the use of optimization is not a special significance of global optimum, but rather its capacity to give mathematical form to the problem itself, and its effectiveness in generating local search algorithms acting on the space of potential solutions to the problem. It is in light of this conclusion that we claim that in data analysis, the following is typically true:

Using optimization methods to solve empirical problems is a form of forcing.

In the remaining part of the present section, we will support this claim through examples. This will allow us to distinguish the two relevant forms of forcing optimality most frequently used in data analysis. We are then left to understand in Sect. 3 and Sect. 4 the reasons of the effectiveness of forcing optimality.

2.1 Forcing optimization methods

In one of the early examples of computational methods inspired by biological systems, Kennedy and Eberhart motivated an innovative optimization method in [22] by assuming that birds move in a flock to optimize their search for food. Modifying some preexisting models of birds flocking behavior [34], they built virtual particles that try to stay cohesive, while individually searching for optimality of a fitness function. The resulting searching method, called particle swarm optimization (PSO), turned out to be extremely popular in solving optimization problems, partly because of the attractiveness of a technique, such as PSO, that does not require much effort to be adapted to specific problems.

Here we want to show how the structure of this technique relies heavily on heuristic rules, to the point that PSO loses its original goal of finding a global optimum of a fitness function. PSO methods starts by setting (possibly randomly) position and velocity of multiple virtual particles, each of them associated with the evaluation of the fitness function at its position. The position and velocity of each particle is then slightly modified, taking into consideration both the location of its best evaluation up to that moment, and the location of the best evaluation of a set of neighboring particles. This updating process is justified by the hope that, as the particles move in the domain of the fitness function, and communicate their respective evaluations with each other, at least one of them will eventually settle its position to the sought for location of the global minimum. To which extent this assumption is true will be discussed shortly, after giving some details on the actual implementation of PSO methods.

Consider a swarm composed by N distinct particles, each of which is defined by a position $x_i = x_i(t) \in \mathbb{R}^n$ and a velocity $v_i(t) \in \mathbb{R}^n$, $(i = 1, ..., N)$, depending on the value of a temporal parameter t. Let $f(x)$, with $x \in \mathbb{R}^n$, be a fitness function that we wish to minimize on a domain $\mathscr{S} \subset \mathbb{R}^n$, and let $x_i(0) \in \mathbb{R}^n$ and $v_i(0) \in \mathbb{R}^n$ be the initial position and the initial velocity vector of the i-th particle, respectively.

We define $b_i(t)$ as the best position that the i-th particle achieved up to time t, which means that we have $f(b_i(t)) \leq f(x_i(t'))$ for all $t' \leq t$ and $b_i(t) = x_i(T)$ for some $T \leq t$. Assuming that each particle has a well defined neighborhood $\mathscr{N}_i$ (either a small set of particles within a given distance, or a set of particles that are a priori considered to be close to each other), we also define $l_i(t)$ as the best position within the neighborhood, which means that $f(l_i(t)) \leq f(b_j(t))$ for all $j \in \mathscr{N}_i$ and $l_i(t) = x_{j'}(T)$ for some $T \leq t$ and $j' \in \mathscr{N}_i$. We then update the velocity and position of each particle according to the following rules [22, 14]:

$$v_i(t+1) = wv_i(t) + \varphi_1 U_1(t)(b_i(t) - x_i(t)) + \varphi_2 U_2(t)(l_i(t) - x_i(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

where $w > 0$ is a parameter called 'inertia weight' which gives a measure of the tendency of a particle to keep its direction of motion, φ_1, φ_2 are also positive and are called acceleration coefficients, and U_1, U_2 are random diagonal matrices with

values in the interval $[0,1)$ regenerated at each iteration. The choice of w, φ_1, and φ_2 is crucial to ensure that the algorithm output does not diverge. The whole term $\varphi_1 U_1(t)(b_i(t)-x_i(t))$ is referred to as 'cognitive component', as it quantifies a sort of memory of the particle, that keeps track of the location of its own best past or current evaluation of the fitness function. The effect of this term is to contribute to the velocity vector $v_i(t+1)$ a component that always points to $b_i(t)$, and that, the further $x_i(t)$ is from $b_i(t)$ the larger it is (not withstanding the variability of $U_1(t)$). The term $\varphi_2 U_2(t)(l_i(t)-x_i(t))$ is referred to as 'social component', since it involves a local comparison within a set of neighboring particles, and it gives the particle a tendency to stay close to the location of the best past or current evaluation of the entire set of neighbors. Like the previous term, it contributes to the velocity vector a component that is always pointing in the direction of the best location ever found by the neighbors.

The purpose of local optimization is encoded in the structure of the PSO algorithm through these two components that decide the direction of the movement of the particle. The halting of the algorithm updating the positions and velocities of the particles is usually done either by setting a very large number of iterations before running the algorithm, or by terminating the iterative updating when the velocities of all or most particles get close to zero (presumably because they are close to the global minimum). In both these cases the estimate of the global minimum will be the minimum among the values of the fitness function at the location of the particles. The use of several particles allows to explore widely the fitness function on the domain $\mathscr{S}$, while maintaining a certain coherence among their evolution. However, there is no evidence in general that running the PSO algorithm will indeed lead to some global optimum in $\mathscr{S}$.

While there are many variants of PSO methods, they broadly respect the structure sketched above, and they turn out to be effective on a large variety of optimization problems, including many in image and signal processing, and in control theory [33]. Here we broadly summarize an image processing application that approaches the issue of image denoising using PSO methods [4]. This is a particularly interesting example because it will introduce a way to build fitness functions that we will further explore in Sect. 2.2.

Let I be a discrete image, given as a real valued function defined on a grid of N discrete points. It is often more useful to represent I as a weighted sum of simpler images, that encode some specific characteristics we may expect in the image, for example, such simpler images could be oscillating patterns with fixed frequency, or they could be sharp spikes that are nonzero only over a few points on the grid. Let us call such collection of images a dictionary $\mathscr{G} = \{g_1, \ldots g_M\}$, and assume that $I = \sum_{m=1}^{M} c_m g_m$ for some coefficients c_m, so that, in a given dictionary $\mathscr{G}$, our image can be represented by the list of coefficients $c = \{c_1, \ldots c_M\}$. We note that in general we need more dictionary elements than the size of the image, i.e. $M > N$ (see [23], chapter 5). In practice the image could be contaminated by noise, and in an idealized scenario we can assume that such noise is additive, that is, we have access only to the noisy image $\tilde{I} = I + W$, where W is a set of values drawn from some probability

distribution. Under these conditions, the dictionary coefficients that we have access to are those of the noisy image $\tilde{I}$, i.e. $\tilde{c} = \{\tilde{c}_1, ...\tilde{c}_M\}$.

How can we reconstruct the original image from these noisy coefficients? It turns out that, if the dictionary is well chosen, we can set to zero or shrink the smallest coefficients (threshold them) without losing the most important characteristics of the original image, for example its edges and texture. Wavelet representations are among the dictionaries that allow this thresholding operation [23]. Without going into the details of their structure, we can say that they entail a way to decompose the image that encodes information at large scales (say, the rough contours of the objects in the image) and information at much finer scales (information on local textures and noise). Then it is possible to build a threshold function $T(\tilde{c}_m)$ that, according to the magnitude of each $\tilde{c}_m$, decides whether that coefficient needs to be replaced by zero, or, possibly, by some other value smaller than $\tilde{c}_m$. For example, early in the development of this field, it was proposed the threshold function defined by $T(\tilde{c}_m) = 0$ if $|\tilde{c}_m| < C$, $T(\tilde{c}_m) = \tilde{c}_m$ if $|\tilde{c}_m| > C$, for a carefully chosen value of C [12, 13]. This turned out to be too crude in some applications where more information is available on the type of images of interest, and more complex threshold functions, depending non linearly on several extra parameters, were suggested. In particular one recent type, developed in [30] and used in [4], depends from three parameters p,q,r. We denote it here by $T(\tilde{c}_m, p, q, r)$. This threshold function can be seen, for each choice of parameters p,q,r, as a piecewise smooth function very close to zero in a symmetric neighborhood close to the origin, and almost linear outside that neighborhood.

The parameters in $T(\tilde{c}_m, p, q, r)$ are chosen to keep the new reconstructed image as close as possible to the original one, and one way to do that is to assume that the error (fitness) function $E(p,q,r) = \|I - \sum_{m=1}^{M} T(\tilde{c}_m, p, q, r) g_m\|_2^2$, is minimized, where we denote by $\|*\|_2^2$ the square of the L_2 norm of a discrete image (i.e. the sum of the squares of the values of the image at each of the N points on which it is defined). Alternatively, under some restrictions on the type of dictionary used, one could minimize the simpler error function $E(p,q,r) = \sum_{m=1}^{M} |T(\tilde{c}_m, p, q, r) - c_m|^2$, as it is done in [4]. Assuming we know one representative image with and without noise, it is possible to minimize the function $E(p,q,r)$ above, with the hope that, once optimal parameters p,q,r are found, the threshold $T(\tilde{c}_m, p, q, r)$ with that choice of parameters will effectively denoise other noisy images.

Now, the complex dependance of the threshold function from its parameters causes a proliferation of local minima for the error function E. Having several local minima makes it very difficult to have a good, single initial estimate for the approximate location of the minimum, that is, a selection of values p_0, q_0, r_0 that are close enough to the globally optimal values. Such good estimate would be necessary if the global minimum on the domain $\mathscr{S}$ is searched with more traditional gradient descent methods ([1], Sect. 7.3), which essentially modify the location of an evaluation of a fitness function in the direction of the steepest negative change of the function. What is more troubling, these methods depend on an estimation of the derivative of the error fitness function, which, in turn, depends from the coefficients

c_m of the original image (with potentially very wide variations) making a numerical estimate of the derivative difficult.

PSO methods can overcome these difficulties thanks to the following two properties: they do not need a good initial estimate for the global minimum, since an entire population of initial evaluations is used instead; they do not need the computation of the derivatives of the error fitness function, and therefore their performance is robust with respect to variations in the image. In fact, in [4] it is shown that PSO methods can be used successfully and quickly to minimize the error function E, either in the basic setting of uniform threshold across all the dictionary, or even the more computationally intensive sub-band setting, where the parameters are chosen independently for subgroups of the dictionary elements. The threshold parameters found by PSO are then shown to give effective image reconstructions, when confronted with other denoising techniques.

The preponderance of applications of PSO methods in image processing is not casual, as optimization problems derived in this setting often display the complex dependance from their parameters argued in the image denoising example (see Sect. 2.2). But despite the wide applicability of these methods, they are known to perform poorly when applied to combinatorial optimization problems [33] where the true optimum is one of a finite number of discrete possibilities that, possibly, grows exponentially with respect to the number of the variables involved in the problem. However, these optimization problems are exactly those that are hardest to solve (many combinatorial optimization problems are NP-complete [35]), and for PSO methods to be a significant advance in optimization we would expect them to perform well when applied to these problems. As noted in the conclusion of [33]:

> most people will not care as to whether their new tool is guaranteed to give the absolute best performance on problem. What they want is something simple and reliable. Finally, probably the PSO has, at the moment, in the mind of many people the sort of magical black box flavor that attracted so many researchers to other area of artificial/computational intelligence before.

This black box belief was briefly discussed in [27] when we suggested that the microarray paradigm has a counterpart, in the use of computational processes, in the belief that a sufficiently complex machine should be able to solve any scientific problem.

The work in [32] suggests that PSO methods can be conceptually and effectively simplified by recognizing the heuristic nature of the social and cognitive components in their implementation. Under this interpretation, they are methods of space exploration whose implementation is guided by a fitness function (to be optimized). However, they do not use in an essential way the analytical properties of the fitness function (for example its derivatives) and they are not expected to eventually converge to a global minimum. It is argued instead that only the actual performance of PSO methods on benchmark problems, where there is a consensus of what constitutes a solution, can discriminate those variants that are faster and more robust.

This means that the effectiveness of PSO methods does not depend on their having been conceived as optimization techniques. In fact, what counts here is not the optimization, but the capacity of the algorithm to appropriately modify the relevant

values (position and velocity of each particle). Even if a fitness function f needs to be appealed to, for the algorithm to work, once this is done, optimization is no more a central concern of the resulting recursive algorithmic procedure. A PSO algorithm is accepted as suitable simply as long as, by following it, some solution to the original empirical problem is found, in a way that does not exceed some predetermined time and resources constraints, rather than in the best possible way.

It is true that a choice of a fitness function to be used by PSO methods might not be effective for a specific problem, and this would require it to be replaced by another function, until an appropriate fitness function is chosen. Still, this does not imply that we have a trial and error procedure, or some sort of hypothetico-deductive approach. PSO methods are shown to be effective, in a very specific algorithmic form, for entire classes of very different empirical problems even though they may not truly optimize sufficiently complex fitness functions. What happens is that a fixed mathematical technique, or better, a well specified algorithm, is forced onto the problem; if trials and errors occur, it is at the level of the fitness function, and always within the framework of PSO methods, which is mostly fixed under the passage from each trial to another. This is not to say that the fitness function itself is never forced onto the problem. Rather the contrary is often true, both in the application of PSO, and in other optimization methods used in data analysis: the specific choice of the fitness function (and its corresponding extrema) does not hinge on a structured heuristic insight into the empirical problem, which prefigures another sort of forcing. This is what we shall see in the next section.

2.2 Forcing fitness functions

The way image denoising is phrased as an optimization problem in Section 2.1 is exemplary of the way in which an optimization framework is used to phrase other complex problems about natural phenomena, irrespective of the specific method used to solve the resulting optimization problem. The emphasis in Sect. 2.1 was on how to solve an optimization problem, already derived from an empirical problem. Here instead we explore, in the context of more sophisticated image processing problems, the difficulties in the preliminary process of translating the empirical problem into an optimization one by choosing a fitness function.

A first consideration in image processing is that a natural image has important features that are relevant to its recognition and classification, for example overall contours, or different uniform textures in different regions, or different objects superimposed to each other. Capturing this structure involves first of all having a compact representation for the image, for dictionaries that efficiently encode edge information ([23], chapter 9), this requirement can be approximately satisfied by requiring the minimization of $E_1 = \sum_{m=1}^{M} |T(\tilde{c}_m)|$, where by $T(\tilde{c}_m)$ we indicate here symbolically the coefficients of the reconstructed image (without being concerned with the analytical shape of the function T), and where we enforce that $\tilde{I} = \sum_{m=1}^{M} T(\tilde{c}_m) g_m$. Minimizing the function E_1 has been shown to lead to a sparse,

compact, representation for the reconstructed image where only a few coefficients are nonzero [12, 13]. Suppose now that we wish to have a sparse image reconstruction and a denoising of the image at the same time. A way to proceed would be to look at the minimization of the overall fitness function $E = \lambda E_1 + E_2$, where $E_2 = \|\tilde{I} - \sum_{m=1}^{M} T(\tilde{c}_m) g_m\|_2^2$ (a modification of the fitness function seen in Sect. 2.1 with respect to the noisy image $\tilde{I}$) and $\lambda > 0$ determines the relative strength of the two fitness terms in E [11]. What is important for our discourse, is that there is now an entire family of fitness functions, parameterized by λ. While under certain restrictive conditions on the dictionary it is possible to select heuristically a specific good value of λ [9], this is not the case in general, and we are left with the problem of choosing which, if any, of the fitness functions is truly the most suitable for the specific noisy image to represent sparsely.

More sophisticated image processing problems are associated to ever more complex fitness functions. Consider for example the problem of identifying incomplete contours of objects in an image. One of the most efficient method to solve it goes under the name of 'active snakes' [21]. These algorithms are based on the idea of superimposing contours (snakes) on the image, and of subjecting these snake contours to some internal forces (minimizing the complexity of the shape of the contours) and some external forces (that maximize the fit of the contours on features of the image). Starting from some initial snake contour guess, these forces slowly change the shape of the snake on the image until it stabilizes on some significant feature within the image (usually the actual contour of an object).

The effect on the snake of internal and external forces is determined by a fitness function depending on several parameters that encode the relative strength of the forces, and to each choice of parameters (to adjust heuristically on the specific problem) corresponds a distinct fitness function. Effectively, we have an entire family of fitness functions, all suitable in principle to solve the problem of finding contours of objects in a given image. However, the dynamics of the snake on the image can be very complex, and different choices of fitness functions lead the snakes to stabilize on very different contours.

The tentative ways in which image processing problems translate into difficult optimization problems makes it clear that fitness functions are often *ad hoc*, and several parameters and fitness terms constraining the optimal solution need to be determined to make them plausible candidates to solve them. The significance of any specific fitness function is weakened, even though we need to have one such function to apply optimization methods. This is in line with the general fact that any application of forcing reduces the significance of our structural understanding of a phenomenon and a related problem. Which of these potentially distinct, but all equally plausible, fitness functions should we optimize? As Brandt notes in [6]:

> this combination of penalty terms creates a monstrous minimization problem [...] It is extremely difficult to solve – *and unnecessarily so.*

The difficulty that Brandt highlights is due to the fact that for a general fitness function F there are no closed, analytical methods to solve the related optimization problem. The only way to look for a solution is often a local search, what he calls

'point by point minimization', an iterative method that, given a point x_i, gives as output a point x_{i+1} in a neighborhood of x_i such that $F(x_{i+1}) \leq F(x_i)$. The iterative process is started by selecting an initial guess x_0 for the potential solution and it is terminated when the distance $|x_{i+1} - x_i|$ between successive outputs of the process is smaller than some preset small $\delta > 0$, in which case we say that the point by point minimization has converged.

We have seen this local approach to optimization already in PSO methods, and indeed for all fitness functions used in the image processing problems above the solution is usually found with one of several recursive local optimization methods, such as gradient descent and Euler-Lagrange methods [21, 8], or interior point methods [11, 5]. We suggest that for a sufficiently complex optimization problem and related fitness functions, arising from any empirical problem, a computational method that attempts to solve the problem is generally bound to be some type of point by point minimization. In particular, for a general fitness function, a point by point optimization process may not converge at all, may not converge quickly enough, or may converge to a point that does not minimize the fitness function on its domain; this limitation is intrinsic to these methods, rather than being incidental to some of the specific examples we have seen. Even more important, there is no guarantee that choosing slightly different fitness functions, with different parameters, will lead to similar solutions. In the next section we will see how "unnecessarily" difficult optimization problems and fitness functions can be reinterpreted and simplified by looking closely at the structure of point by point minimization processes.

3 Brandt's principle

In this section we suggest that both PSO and the other methods developed by forcing optimization onto empirical problems are governed by one methodological principle, which we call 'Brandt's principle' (derived and generalized from considerations of Achi Brandt in [6]) which clarifies the role and appropriate usage of point by point minimization. We argue moreover that this principle shows a specific, operative way in which the link between structural understanding of a phenomenon and successful problem solving through data analysis can be broken, without apparent impact on the effectiveness of the problem solving itself. In particular, we show that the actual implementation strategies of Brandt's principle, while depending on partial, fragmented information on the phenomenon, are an expression of the microarray paradigm.

The starting point of our analysis is the strategy suggested in [6, 7] to solve image processing problems. Those reviews show how direct multiscale algorithms, that satisfy a whole set of contrasting criteria at different stages, by focusing on different scales at each of these stages, can perform as well as global regularization techniques in finding solutions of some ill posed problems (i.e. problems where the solution is not uniquely defined, and/or such that the solution is not robust under variations of the input of the problem). The relevance of multiscale methods is not

surprising for image analysis, given what we have said in Sect. 2.1 concerning the way information about the relevant image are encoded differently at different scale of detail (contours versus texture for example). The main focus and interest in [6, 7] is to understand how to approach optimization problems pertaining to image processing and to partial differential equations with multiscale methods, and how to build, case by case, such methods. Here we would like to expand the scope of this insight to wider classes of problems and methods.

According to Brandt, in many cases, the search for optimization can be replaced by "a more universal and often far easier" approach, which just consists in admitting:

> a solution which is just the end product of a suitable *numerical process* not necessarily designed to satisfy, even approximately, any one governing criterion. ([6], page 60)

The problem with this highly agnostic approach (in our sense) is how to identify suitable numerical processes. A criterion Brandt suggests for this purpose is this:

> the amount of computational work [in a numerical process] should be proportional to the amount of real physical changes in the computed system. Stalling numerical processes must be wrong. [7]

It would follow that what truly distinguishes a solution of a problem, or decides that the time has come to switch to a different solving algorithm, is a near-steady state of the algorithm output over successive iterations of the algorithm itself; stalling algorithms are wrong either because we have found a solution and we should terminate the algorithm (as it is the case for point by point optimization algorithms that have converged), or because we need to switch to a new algorithm. We state succinctly the radical idea that underlies the two quotations above in the following principle, which we name after Achi Brandt. And we claim that its domain of applicability is far more encompassing than the field of multiscale methods, and that it can be taken as the organizing principle of most data-driven computational methods:

Brandt's principle: An algorithm that approaches a steady state in its output has found a solution to a problem, or needs to be replaced.

This principle proposes a fundamental shift away from the search of best solutions to problems. This shift underscores the data-driven nature of the methods in data analysis, and complements it. Since Brandt's principle does not have a built-in criterion to establish when a solution has been found (instead of having reached a stage in which one has to switch from an algorithm to another), the output of the relevant numerical process is to be checked, by external validation means, for its usefulness: a fuzzy concept in general, which nevertheless does have quite precise renderings in many specific applications. For many empirical problems, a precise task has to be achieved, and any method that can achieve such a task would be suitable. Think, for example, of the problem of having an autonomous, driverless car travel from one location to another in rough terrain; achieving this task, in any way,

is the really interesting problem, because of the dangers of the car getting overtopped, or otherwise incapacitated. Finding the very best route in terms of shortest distance, or fuel efficiency is secondary. In general, the distinction we make here is between optimality, which usually designate some singular points in the space of solutions, and usefulness, which rather designate entire regions in the space of solutions.

While Brandt's principle deemphasizes the search for optimal solutions to problems, it is satisfied by local, point by point optimization methods. We have already seen in Sect. 2 that wide classes of computational methods, including PSO and image processing methods, can ultimately be expressed in this way. Moreover, most data analysis methods attempt to fit a model on data by minimizing some suitable error function [19]. Because the fitness functions that correspond to such methods are usually both *ad hoc* and complex, local minimization techniques are preferred in this context. This is true in particular for artificial neural networks, currently some of the most powerful data analysis techniques ([20], chapter 10). As we pointed out in [28], the success of artificial neural networks can be ascribed in great part to the effectiveness of a particular local recursive process to estimate their parameters, the backpropagation algorithm. In turn, this algorithm is essentially a form of gradient descent algorithm ([19], Sect. 11.4).

In light of the pervasiveness of local optimization techniques in data analysis, we can therefore claim that most of its methods satisfy Brandt's principle when implemented in an effective algorithmic form. Moreover, some of the most important types of classification methods, such as committee machines ([20] chapter 14) can be shown to satisfy Brandt's principle without an explicit local optimization implementation. This is particularly significant when we consider that boosting algorithms, singled out in [26] as one of the most significant embodiments of the microarray paradigm, are a type of committee machines ([20], Sect. 14.3). In fact, committee machines are built starting from more basic, structurally similar classification algorithms (classifiers). A standard committee machine switches among these basic classifiers, each attempting to find a solution for a slightly different version of some initial classification problem, and it eventually combines them into a single, potentially more accurate classifier (we refer to [26] for a general methodological analysis of classification problems and classifiers). Crucially, the decision of switching among the basic classifiers is made on the basis of the convergence to a stable tentative solution to the classification problem at each iteration of the process. A similar, sequential approach is also used in hybrid classification methods [10]. The difference with respect to committee machines is that the basic classifiers utilized by hybrid methods do not necessarily share a common structure.

We did not emphasize so far the aspects of an algorithm that are uniquely related to a specific problem. Brandt talks about "suitable" algorithms, but what does it mean in the context of a specific problem? To begin to answer this question, we recall that, in Sect. 2.2, the fitness functions for image processing problems are built piece by piece on the basis of partial assumptions on the images. While individually these assumptions do encode some understanding of the structure of images, the resulting fitness functions, with all their parameters to be chosen, do not offer a

transparent way to interpret the solution to a given problem, and their complexity does not generally allow for a convergence to the optimum. Fitness functions are forced on the problem and they can, at best, be justified as rough quantitative *a priori* constraints on the form of a solution.

More generally, in image processing the *a priori* constraints are our assumptions on what constitutes a natural image or a contour, and these assumptions affect the choice of image dictionaries. It is possible however to make this choice, and establish our assumptions on images, automatically, for example by allowing sparsity to be the governing criterium of image representation, and using large databases of natural images to search for the best dictionary elements that, on average, optimize sparsity on a given image of the database [31, 17]. This process is computationally intensive, but it can be achieved in many cases, and the resulting choices of dictionaries can then be used for further image processing problems. Of course, the assumption of sparsity is not justified from the data per se, but is required by our limited resources.

We note also that the use of specific prior assumptions on a phenomenon to force optimization is similar to the basic approach of pattern theory [18] and Bayesian analysis [3] (other very powerful theories to model and classify complex phenomena), where the solution to empirical problems is contingent on the identification of appropriate priors on the probability distributions of the stochastic processes used to model the underlying phenomena.

From this analysis, we can conclude that the building blocks of methods that respect Brandt's principle are based on the unavoidable inclusion of partial pieces of prior information (what, following a customary usage, we can call 'priors', *tout court*) on a phenomenon. This is an important point that we need to clarify, as apparently the presence of priors proper to a particular phenomenon may be seen to be in contradiction with the microarray paradigm, and the whole agnostic approach to data analysis. One could argue indeed that, if for each problem we should identify the exact prior information that is necessary to solve it, we would have no reason to say that the microarray paradigm is at work.

Still, as we have seen in Sect. 2.1, even the use of specific priors (fitness functions in that case) does not guarantee that we can understand the reason a specific output of an optimization method solves the relevant empirical problem. If this is not the case, and it happens very often, the problem solving is still agnostic in our sense, and the microarray paradigm applies. In other terms, the fact that some priors are proper to a certain problem does not mean that their identification depends on a structural understanding of the subjacent phenomenon.

In data analysis we assume a huge quantity of data about a phenomenon, and from these data we can extract many significant features of the phenomenon, in an entirely data driven way if necessary, to the effect that these features are generally not structurally related to each other. More than that, it is often quite hard to guess in advance which subgroup of the features so detected will be useful to solve a given problem. This large number of priors can be partially structured, for example in the form of plausible guesses on the mathematical nature of the data (as in Bayesian analysis, where we may have information on the shape of probability distributions

about the phenomenon), or it can be simply in the form of large collection of raw data from the phenomenon. In both cases, it is the sheer size of priors that set us squarely within the microarray paradigm: we need as many priors as possible to solve a problem in a data driven way, and to have an excess of partially structured priors is as opaque and agnostic as to have none.

We finally note that effective hybrid classification methods differ significantly in the order and the type of the basic classifiers they use. Similarly, image processing methods based on optimization allow for significantly different, equally suitable fitness functions. This suggests that, when building a method based on Brandt's principle, the specific way the priors are put together is not essential: what counts is the ability to distinguish when an algorithm produces dynamically changing outputs, and when instead it gives nearly steady outputs. And the reason for the usefulness of the alternating process advocated by Brandt's principle is that it does not explore the solution space randomly, but according to priors that, no matter how they are combined, encode preferential ways to look at the problem itself. Therefore the expectations on algorithms governed by Brandt's principle are the following: they have to be easy to build, starting from a large number of weak assumptions on a phenomenon and a problem about it; their nearly steady outputs have to be exceptional and clearly identifiable; and switching among steady states outputs has to provide a fast exploration of the solution space of the problem at hand.

4 Data analysis of historical phenomena

We have seen in Sect. 3 how the process of forcing optimality and its effectiveness can be reinterpreted in light of Brandt's principle, and we have argued that this principle governs most agnostic, data-driven methods. In this Section we explore the appropriateness of using such methods, consistent with Brandt's principle, to solve problems about historical phenomena (introduced in [29]). We perform this analysis by looking at one of the organizing principles that have been suggested in recent years to make sense of biological processes, the so called principle of developmental inertia [25]. We will show the generality of developmental inertia as an organizing principle of historical phenomena, subject to morphogenesis (a term used here in the general and original meaning of change of characteristics). Finally we will draw parallels between Brandt's principle and the principle of developmental inertia to find evidence of the relevance of the former (and of data analysis in general) for the solution of problems about historical phenomena.

In [29] we suggested that biology and social sciences could be preferred domains of exploration of data analysis, as sciences concerned with "historical phenomena", i.e. phenomena significantly constrained by their past, historical development. We briefly argued moreover that it is possible to describe historical phenomena as "those phenomena whose development can only be constrained locally (in time and/or space) by (potentially multiple) optimization processes acting on subsets of variables, and in such a way that the functions to be optimized change over

long periods of time" [29]. This characterization was motivated by an analysis of fitness landscapes in evolutionary biology. There are important relations between these phenomena and Brandt's principle, in particular in the way the latter allows to reinterpret point by point minimization algorithmic processes. To understand these relations, we first explore the general structure of historical phenomena, starting with ideas from developmental biology.

In [25] Alessandro Minelli, in order to conceptualize the way organisms structure themselves, suggests the *principle of developmental inertia*; generally stated, it asserts that we can see:

> [biological] developmental processes...as deviations from *local self-perpetuation of cell-level dynamics* ([25], page 119).

Similarly to the concept of inertial conditions in physics, the main purpose of the notion of developmental inertia is to identify an appropriate singular state in the relevant processes, the null (or inertial) state of these processes, this is the state that the system would stay in, if no perturbations (external to this very state) had occurred. The intrinsic nature of such null state is less relevant than the nature of the perturbation acting on it, allowing the system to evolve along different lines. It is argued in [25] that:

> [developmental] inertial conditions [...] *do not represent origin*, in the ordinary meaning of the word, but only a convenient "zero" term of comparison for the study of something that happens in time– a segment of history, without prejudice of what happened before it.

The identification of suitable inertial conditions in biological systems leads to fruitful reinterpretations of asymmetry, segmentation and regeneration in complex full-grown organisms, exactly by pointing out the appropriate, null state of biological development. A particularly simple example can be seen in the context of embryo development, where the appropriate inertial state of embryos is the reproduction, symmetrical in space and indefinite in time, of the same basic tissue structure. However, in a real system, developmental inertia is constantly perturbed ([25], page 123), which allows complex, inhomogeneous and asymmetrical organisms to form.

Developmental inertia may be a key principle in conceptualizing biological developmental processes, but it would need to have much wider applications to be an organizing principle of general historical phenomena. Since social phenomena are arguably, together with biological ones, the most important type of historical phenomena, the principle of developmental inertia should apply to these phenomena as well. Showing case by case the usefulness and validity of this principle in this new context would require a large survey in itself (which cannot be performed here). However, it would appear that, if we look at a social system as a collection of interacting individual agents with their own decision rules for behavior [16], collective agreement among such individuals can be assumed to be an inertial state. For example, social norms have been shown to have a tendency to become entrenched and to be self-enforced [15], to the effect that their entrenchment and their spreading across individuals could be taken, locally in time and space, as an inertial state in a society.

A better, more general route to justify the principle of developmental inertia for social phenomena is to rely on the broad, strong homology of social and biological systems. This homology is made explicit in accounts such as the "living systems theory" detailed in [24], that emphasize the notion of process in all living systems, whether biological or social [2]. Assuming its truth, it is likely that general principles from biology transfer to social phenomena, and we can conclude that the principle of developmental inertia applies to a wide range of historical phenomena, both biological and social in nature.

We also note that, at any given time, there is accumulation of structures in the states of historical phenomena; for example, accumulation of tissues differentiation in the embryo, and accumulation of legacy social norms in societies. This accumulation of structures gives clues about the inertial states that occurred along the history of a phenomenon: we can identify a structure as distinct from the whole (for example a specific tissue in an embryo) only because some inertial state (the repeated proliferation of that tissue) is replaced by some other inertial state (the differentiation into a new tissue). This suggests focusing on those characteristics of an historical phenomenon that can be ascribed to some unperturbed developmental inertial state that occurred in its past. We shall refer to them as the developmental structures of such a phenomenon.

In light of the broad applicability of the principle of developmental inertia to historical phenomena, it is striking how similar this principle is to Brandt's principle. Both principles identify appropriate null states: self-perpetuating cell dynamics for developmental inertia; near-steady state outputs of algorithms for Brandt's principle. Both principles see the breakdown and deviation from a null state as essential: to morphogenesis processes in complex organisms in the case of developmental inertia; to computational processes that solve problems in the case of Brandt's principle. Moreover, to have a near-steady state output of an algorithm implies that its (dynamical) computational processes are nearly repeating themselves, so that we can speak of self-perpetuation of dynamical processes also in the context of Brandt's principle. We believe this strong similarity is fundamental to the understanding of the effectiveness of data analysis, and we highlight it in the following proposition:

The principle of developmental inertia, as applied to historical processes, and Brandt's principle, as applied to computational processes, are homologous.

For historical phenomena, there is no compact description of the totality of developmental structures, so that it is necessary to identify and collect as many individual structures as we can to solve problems, which is in line with the microarray paradigm. Indeed, each developmental structure of an inertial state essentially differs from, and cannot be reduced to, the developmental structure due to another, contemporary or subsequent, inertial state. If it is necessary to focus on distinct developmental structures to solve a problem, these structures have to be identified, and harnessed, independently of each other. The author of [25] talks of distinct centers of developmental dynamics that are "everything everywhere", so that the proliferation

of developmental structure does not reduce itself to a simple, unifying description. However, the collection of all such developmental structures does encode, to some extent, the state of the historical phenomenon.

Computational methods that respect Brandt' principle, or 'Brandt's methods', are uniquely suitable to take advantage of the incoherent proliferation of developmental structures, since each of these structures can be used as a prior for a corresponding algorithm, a building block for Brandt's methods. Alternating among these algorithms, when their output reaches a steady state, may be the preferred way to successfully explore the solution space of a problem. For general problems about historical phenomena, a suitable Brandt's method may be as efficient a way to search for solutions as any other method. The lack of global finality or optimality of developmental structures frustrates any attempt at faster exploration of the solution space.

As we have seen in Sect. 3, Brandt's principle offers a fruitful, theoretical scaffolding for forcing optimization and, more generally, for data analysis. The homology of Brandt's principle with the principle of developmental inertia suggests something more: data analysis, rather than being simply an heuristic, preliminary set of tools, could actually be the privileged way to approach historical phenomena and their problems with quantitative, theoretical tools.

References

1. Arfken, George B. , Hans J. Weber. 2005. *Mathematical Methods for Physicists*. Boston: Elsevier Academic Press.
2. Bailey, Kenneth D. 2006. Living Systems Theory and Social Entropy Theory. *Systems Research and Behavioral Science* 23:291–300.
3. Berger, James O. 2010. *Statistical Decision Theory and Bayesian Analysis.* New York: Springer-Verlag.
4. Bhutada G. G., R. S. Anand, and S. C. Saxena. 2012. PSO-based learning of sub-band adaptive thresholding function for image denoising. *Signal, Image and Video Processing* 6(1):1–7.
5. Boyd, Stephen, Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge, UK ; New York Cambridge University Press.
6. Brandt, Achi. 2001. Multiscale Scientific Computation: Review 2001. In *Multiscale and Multiresolution Methods: Theory and Applications*, eds. Timothy J. Barth, Tony F. Chan, Robert Haimes. Berlin ; New York: Springer Verlag.
7. Brandt, Achi, and Oren E. Livne. 2011. *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*. Philadelphia: Society for Industrial and Applied Mathematics.
8. Bresson, Xavier, Selim Esedoḡlu, Pierre Vandergheynst, Jean-Philippe Thiran, Stanley Osher. 2007. Fast Global Minimization of the Active Contour/Snake Model. *Journal of Mathematical Imaging and Vision* 28(2)151–167.
9. Chen, Scott Shaobing, David L. Donoho, Michael A. Saunders. 2001. Atomic decomposition by basis pursuit. *SIAM Review* 43(1):129–159.
10. Delimata, Pawel, Zbigniew Suraj. 2013. Hybrid Methods in Data Classification and Reduction. In *Rough Sets and Intelligent Systems - Professor Zdzisław Pawlak in Memoriam*, eds. Andrzej Skowron and Zbigniew Suraj. *Intelligent Systems Reference Library* 43:263–291.
11. Donoho, David L., Michael Elad and Vladimir N. Temlyakov. 2006. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inf. Theory* 52:1:6–18.

12. Donoho, David L., Iain M. Johnstone. 1994. Ideal spatial adaptation by wavelet shrinkage. *Biometrika* 81(3):425–455.
13. Donoho, David L., and Iain M. Johnstone. 1995. Adapting to unknown smoothness via wavelet shrinkage. *J. Am. Stat. Assoc.* 90(432):1200–1224.
14. Dorigo, Marco et al. 2008. Particle swarm optimization. In: Scholarpedia, 3(11):1486.
15. Epstein, Joshua M. 2001. Learning to Be Thoughtless: Social Norms and Individual Computation. *Computational Economics* 18:9–24.
16. Epstein, Joshua M. 2006. *Generative Social Science: Studies in Agent-Based Computational Modeling.* Princeton: Princeton University Press.
17. Field, David J. 1999. Wavelets, vision and the statistics of natural scenes. *Phil. Trans. R. Soc. A* 357:2527–2542.
18. Grenander, Ulf, and Michael Miller. 2007. *Pattern Theory: From Representation to Inference*. Oxford: Oxford University Press.
19. Hastie, Trevon, Robert Tibshirani, Jerome Friedman. 2009. *The Elements of Statistical Learning*. New York: Springer.
20. Izenman, Alan J. 2008. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. New York: Springer.
21. Kass, Michael, Andrew Witkin, and Demetri Terzopoulos. 1987. Snakes: Active contour models. *Int. Journal of Computer Vision* 1(4):321–331.
22. Kennedy, James, Russell C. Eberhart. 1995. Particle swarm optimization. In *Proceedings of the IEEE 978 International Conference on Neural Networks, Perth, WA* 4:1942–1948. New York: IEEE.
23. Mallat, Stephane. 2008. *A Wavelet Tour of Signal Processing*. San Diego: Academic Press.
24. Miller, James G. 1978. *Living Systems*. New York: McGraw Hill.
25. Minelli, Alessandro. 2011. A principle of Developmental Inertia. In *Epigenetics: Linking Genotype and Phenotype in Development and Evolution*, eds B. Hallgrimsson and B. K. Hall. Berkeley, CA: University of California Press.
26. Napoletani, Domenico, Marco Panza, and Daniele C. Struppa. 2011. Agnostic science. Towards a philosophy of data analysis. *Foundations of Science* 16(19):1–20.
27. Napoletani, Domenico, Marco Panza, and Daniele C. Struppa. 2013. Artificial diamonds are still diamonds. *Foundations of Science* 18(3):591–594.
28. Napoletani, Domenico, Marco Panza, and Daniele C. Struppa. 2013. Processes rather than descriptions? *Foundations of Science* 18(3):587–590.
29. Napoletani, Domenico, Marco Panza, and Daniele C. Struppa, 2014. Is big data enough? A reflection on the changing role of mathematics in applications. *Notices of the American Mathematical Society* 61(5):485–490.
30. Nasri, Mehdi, Hossain N. Pour. 2009. Image denoising in the wavelet domain using a new adaptive thresholding function. *J. Neurocomput.* 72:1012–1025.
31. Olshausen, Bruno A., and David J. Field. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381:607–609.
32. Pedersen, Magnus E.H., and Andrew J. Chipperfield. 2010. Simplifying Particle Swarm Optimization. *Applied Soft Computing* 10(2):618–628.
33. Poli, Riccardo. 2008. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*. doi:10.1155/2008/685175.
34. Reynolds, Craig W. 1987. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21(4):25–34.
35. Wegener, Ingo. 2005. *Complexity Theory: Exploring the Limits of Efficient Algorithms*. Berlin; New York: Springer.