Deflection and Stretching Techniques for Detection of Multiple Minimizers in Multimodal Optimization Problems

Konstantinos E. Parsopoulos and Michael N. Vrahatis

Abstract Multimodal optimization refers to problems where the detection of many local or global minimizers is desirable. A number of methodologies have been developed in the past decades for this purpose. Deflection and stretching are two techniques that can be integrated with any optimization algorithm in order to detect multiple minimizers by properly transforming the objective function. Requiring only a minimal number of control parameters, both techniques have been used with metaheuristics as well as gradient-based optimization algorithms, enhancing their performance while demanding only minor implementation effort. Up until now their applications span various scientific fields, ranging from game theory and numerical optimization to astrophysics, computational intelligence, and medical informatics. The present chapter offers a comprehensive presentation of the two techniques and demonstrates their use through simple examples. Also, their latest developments and applications of the past two decades are concisely reviewed.

1 Introduction

The ongoing scientific and technological developments produce challenging problems that often involve the optimization of multimodal functions with numerous local and global optimizers. Without loss of generality we will henceforth refer only to minimization cases, while maximization can be straightforwardly addressed by changing the sign of the objective function. Even in relatively simple cases, the multimodal minimization problem can become NP-hard [8]. In several applications of

K. E. Parsopoulos (✉)
Department of Computer Science and Engineering, University of Ioannina,
GR-45110 Ioannina, Greece
e-mail: kostasp@cse.uoi.gr

M. N. Vrahatis
Computational Intelligence Laboratory (CI Lab), Department of Mathematics,
University of Patras, GR-26110 Patras, Greece
e-mail: vrahatis@math.upatras.gr

M. Preuss et al. (eds.), *Metaheuristics for Finding Multiple Solutions*,
Natural Computing Series,
https://doi.org/10.1007/978-3-030-79553-5_6

this type, it is required to detect some or all global and/or local minimizers. A typical example coming from game theory is the detection of Nash equilibria that correspond to steady-state solutions of a game [21]. Different Nash equilibria correspond to different outcomes of the game. Thus, the outcome of the game can be predicted if all Nash equilibria are known. Another interesting application is the detection of periodic orbits of nonlinear mappings, which are used to model conservative or dissipative dynamical systems. Fixed points, i.e., points that remain invariant by the mapping, concentrate scientific interest. Developing techniques for detecting multiple fixed points has been an area of ongoing research for many decades [27].

The past decades have witnessed the development of various efficient and effective optimization algorithms. The introduction of methods aiming at the detection of multiple minimizers either through specialized ad hoc procedures (operators) or by using external techniques integrated with the corresponding algorithms is of utmost importance. *Niching* is a term used in metaheuristics literature to describe methods that maintain multiple solutions in multimodal domains, in contrast to existing evolutionary and swarm intelligence optimization techniques that have been designed to detect single solutions. An excellent review of niching techniques is offered by Li et al. in [10].

The present chapter is devoted to the *deflection* and *stretching* techniques [20], which were introduced as mathematical tools for addressing two different problems in multimodal optimization. Specifically, deflection is designed to facilitate the detection of many (local or global) minimizers, while stretching aims at concurrently alleviating many local minimizers. Both techniques are based on the filled functions approach where the original objective function is transformed after the detection of a new minimizer. The applied transformation aims at eliminating the targeted minimizer (and possibly some other minimizers) by transforming it to a maximizer.

Requiring only a minimal number of control parameters and minor implementation effort, both deflection and stretching have been successfully used with metaheuristics as well as gradient-based optimization algorithms. Up until now their applications span various scientific fields, ranging from game theory and numerical optimization to astrophysics, computational intelligence, and medical informatics.

The rest of the chapter is organized as follows: Sect. 2 is devoted to deflection and its applications, while Sect. 3 analyzes stretching and presents a number of recent variants and applications. Indicative experimental results are offered for both techniques in Sect. 4. The paper closes with conclusions in Sect. 5.

2 Deflection Technique

In the following paragraphs, the basic deflection scheme is presented and demonstrated on a well-known test function, followed by a review of recent applications.

2.1 Basic Scheme

The deflection technique has been studied in Magoulas et al. [13] for the alleviation of local minima in artificial neural networks training with the backpropagation algorithm. Later it was adopted for the detection of multiple minimizers with the particle swarm optimization algorithm by Parsopoulos and Vrahatis [19, 20]. It belongs to the category of filled functions techniques [5, 6], where the original objective function is transformed into a new one where the already detected minimizers are eliminated. This way, optimization algorithms are driven away from the detected minimizers, prohibiting their repetitive detection in subsequent runs.

Putting it formally, let:

$$f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}, \tag{1}$$

be the original objective function under consideration. Let us make the necessary assumption that $f(\mathbf{x})$ is bounded from below, and let:

$$M = \left\{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_m^*\right\},$$

be a set comprising already detected minimizers. Then, deflection replaces $f(\mathbf{x})$ with a transformation of the following type:

$$F_M(\mathbf{x}) = \frac{f(\mathbf{x})}{T_1(\mathbf{x}; \mathbf{x}_1^*, \lambda_1) T_2(\mathbf{x}; \mathbf{x}_2^*, \lambda_2) \cdots T_m(\mathbf{x}; \mathbf{x}_m^*, \lambda_m)}, \tag{2}$$

where $\mathbf{x}_i^* \in M$ for all $i = 1, 2, \ldots, m$, and $T_i(\mathbf{x}; \mathbf{x}_i^*, \lambda_i)$ are proper functions with control parameters $\lambda_i \in \mathbb{R}$, respectively. These functions shall be ideally selected such that $F_M(\mathbf{x})$ has exactly the same minimizers as $f(\mathbf{x})$ except for the already detected ones in M. This property is called the *deflection property* [13] and dictates that any sequence of points converging to a detected minimizer:

$$\lim_{k \to \infty} \mathbf{x}_k = \mathbf{x}_i^*, \quad \mathbf{x}_i^* \in M,$$

shall not produce a minimum of $F_M(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}_i^*$. It shall be noticed that the product in the denominator of Eq. (2) consists of one term for each detected minimizer. Thus, the optimization algorithm initially starts with the set $M = \emptyset$, and the original function $f(\mathbf{x})$. Then, as soon as a minimizer is detected, the corresponding term is multiplied with the current denominator, producing a new transformation that is subsequently used. Thus, the form of the deflection transformation dynamically changes during the run of the algorithm, depending on the detected minimizers.

A critical point is the magnitude of the transformation induced by each function $T_i(\mathbf{x}; \mathbf{x}_i^*, \lambda_i)$. Although it is desirable to eliminate $\mathbf{x}_i^*$ from the set of minimizers of $F_M(\mathbf{x})$, it is equally important to confine the effect of the transformation only to a neighborhood of $\mathbf{x}_i^*$ that ideally covers only its basin of attraction. For this reason, the use of tunable functions $T_i(\mathbf{x}; \mathbf{x}_i^*, \lambda_i)$ is necessary.

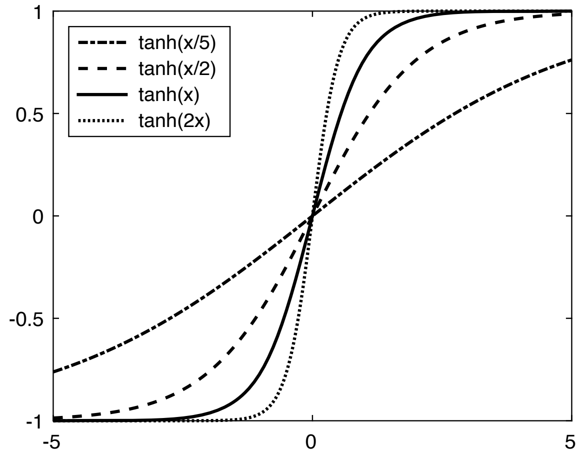

Fig. 1 Plot of the $\tanh(\lambda x)$ function for different values of $\lambda > 0$ and $x \in [-5, 5]$

As proposed in [19], the hyperbolic functions:

$$T_i(\mathbf{x}; \mathbf{x}_i^*, \lambda_i) = \tanh(\lambda_i \|\mathbf{x} - \mathbf{x}_i^*\|), \quad \lambda_i > 0, \tag{3}$$

fulfill the aforementioned properties, with $\|\cdot\|$ denoting a distance norm (typically the Euclidean ℓ_2-norm). Figure 1 illustrates the shape of the $\tanh(\lambda x)$ function for different values of its parameter. Focusing on the nonnegative values of x, we can clearly see that the function is equal to zero for $x = 0$, while it asymptotically converges to 1 for higher values. Therefore, the effect of $T_i(\mathbf{x}; \mathbf{x}_i^*, \lambda_i)$ of Eq. (3) in the deflection transformation $F_M(\mathbf{x})$ of Eq. (2) is expected to be huge when $\mathbf{x}$ lies close to $\mathbf{x}_i^*$ because, as $\|\mathbf{x} - \mathbf{x}_i^*\|$ approximates zero, it produces a singularity point of $F_M(\mathbf{x})$ at $\mathbf{x}_i^*$. On the other hand, the effect becomes milder as $\mathbf{x}$ moves away from $\mathbf{x}_i^*$. The magnitude of the transformation is essentially controlled by the parameter λ.

It can be easily derived from Eq. (2) that the sign of $f(\mathbf{x})$ plays crucial role. Indeed, the proposed deflection approach can work properly only if $f(\mathbf{x}) > 0$ for all $\mathbf{x} \in S$. For this reason, we will henceforth assume that this property holds by definition for the problem under consideration. In cases where this property does not hold or it is unclear whether the objective function is strictly positive everywhere in the search space, the user can enforce it by simply using a shift-up transformation of $f(\mathbf{x})$ as follows:

$$f_\alpha(\mathbf{x}) = f(\mathbf{x}) + \alpha, \tag{4}$$

where $\alpha > 0$ is a sufficiently large positive constant. Possible estimation (e.g., through Monte Carlo sampling) of a lower bound of $f(\mathbf{x})$ can be very useful to this end. Otherwise, arbitrarily large positive values of α can be used.

Let us now demonstrate the deflection technique on a well-known 2-dimensional test problem, namely, the Levy no. 5 test function, defined as:

$$f(\mathbf{x}) = \zeta_1(\mathbf{x})\,\zeta_2(\mathbf{x}) + (x_1 + 1.42513)^2 + (x_2 + 0.80032)^2, \tag{5}$$

where:

$$\zeta_1(\mathbf{x}) = \sum_{i=1}^{5} i\ \cos((i-1)x_1 + i), \qquad \zeta_2(\mathbf{x}) = \sum_{j=1}^{5} j\ \cos((j+1)x_2 + j).$$

In the range $[-10, 10]^2$ the Levy no. 5 test function has 760 local minima. We now focus on the range $[-2, 2]^2$, which is illustrated in Fig. 2. The global minimizer of the function in this range is $\mathbf{x}^* = (-1.3069, -1.4248)^T$, denoted with a black star in the lower left basin of the contour plot of Fig. 2, with $f(\mathbf{x}^*) = -176.1376$. Since this objective function is not always positive, we transformed it according to Eq. (4) with $\alpha = 180$. Also, we shall note that its local minima become deeper as we approximate the global minimizer.

In order to demonstrate the deflection technique, we selected a number of minimizers to apply the deflection transformation. Figure 3 illustrates the deflection transformation applied on the local minimizer $\mathbf{x}_1 = (-0.3521, -0.8003)^T$ with $f(\mathbf{x}_1) = -144.3250$ for $\lambda = 1$ (up) and $\lambda = 2$ (down). Apparently, the effect of the deflection transformation is ameliorated as λ increases, while smaller values have wider influence affecting also neighboring maxima. Figure 4 illustrates the concurrent application of deflection on the three local minimizers $\mathbf{x}_1 = (-0.3521, -0.8003)^T$, $\mathbf{x}_2 = (-1.3069, -0.1957)^T$, and $\mathbf{x}_3 = (-0.3557, 0.3330)^T$, for $\lambda = 2$ (up) and $\lambda = 1$ (down). Deflection applied on multiple minimizers becomes increasingly beneficial because it replaces the detected minimizers with maximizers that can repel the minimization algorithms toward undiscovered minimizers.

A critical observation is that higher values of λ may have only local effect but they tend to introduce new local minima in the neighborhood of the deflected point. This is the well-known *mexican hat effect*; addressing it remains an open problem when the basins of attraction of the minimizers are unknown. Nevertheless, the newly introduced local minima are always higher than the deflected ones and, thus, they can be explicitly neglected by a stochastic algorithm. This can be achieved by using the *repulsion* technique proposed and integrated with deflection in [19] for the particle swarm optimization algorithm.

According to this approach, whenever a particle $\mathbf{x}$ moves into the repulsion area of a deflected minimizer $\mathbf{x}_i^*$, which is defined as a sphere of radius $\rho_i > 0$ around the minimizer, it is immediately repelled away by assuming the new position:

$$\mathbf{x} := \mathbf{x} + \rho_i \frac{\mathbf{x} - \mathbf{x}_i^*}{\|\mathbf{x} - \mathbf{x}_i^*\|}.$$

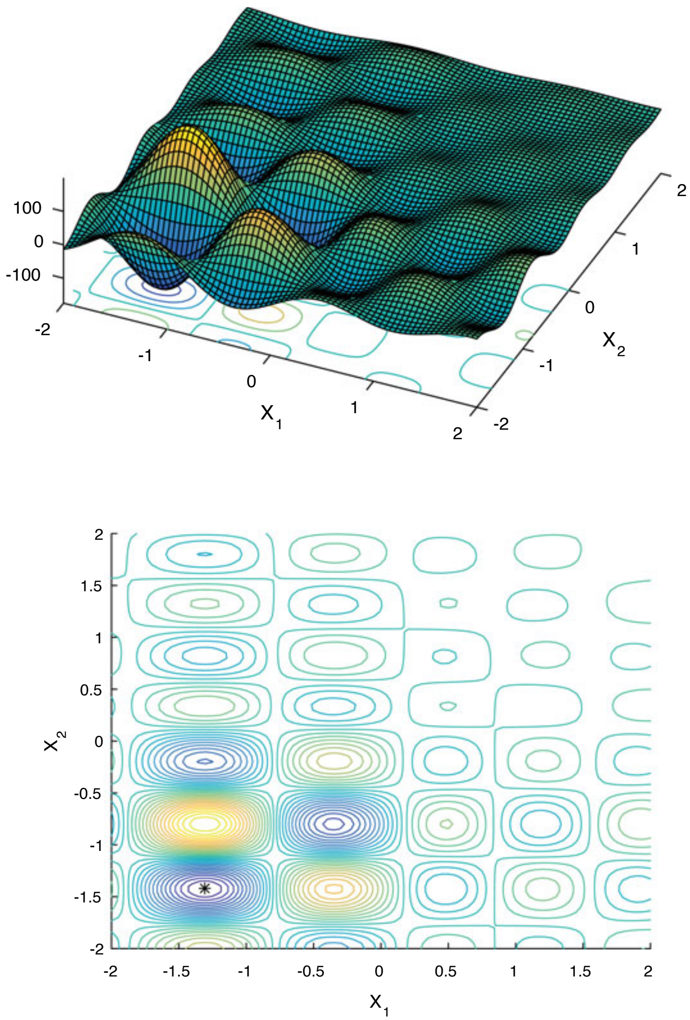

Fig. 2 Surface plot (up) and contour plot (down) of the Levy no. 5 test function of Eq. (5) in the range $[-2, 2]^2$

Again, the problem is the determination of ρ_i, which shall be large enough to prohibit the algorithm from converging close to $\mathbf{x}_i^*$ but, at the same time, avoiding enclosing other minimizers in its repulsion area. Possible knowledge regarding the distribution of the minimizers in the search space may be beneficial for a proper setting. Otherwise, small fixed values of ρ_i are preferable.

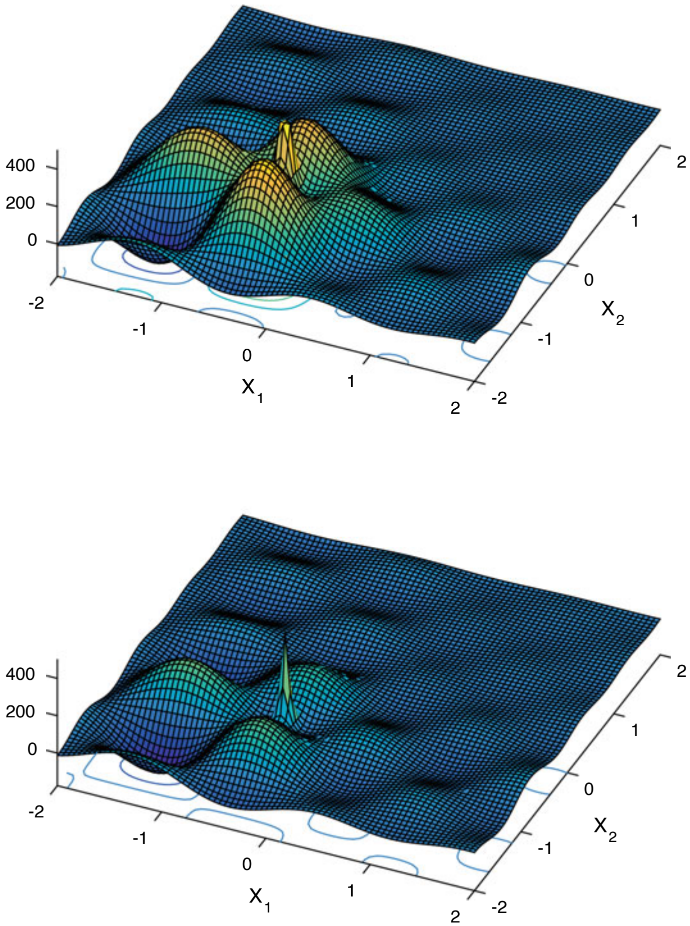

Fig. 3 Deflection transformation applied on the local minimizer $\mathbf{x}_1 = (-0.3521, -0.8003)^T$ for $\lambda = 1$ (up) and $\lambda = 2$ (down)

2.2 Variants and Applications

The deflection technique proved to be beneficial for various algorithms in diverse scientific fields. Important applications have appeared in computational intelligence, game theory, and astrophysics, including optimization, partitioning, Nash equilibria, and periodic orbit problems.

The alleviation of local minima in artificial neural network training was the inaugural application of deflection in Magoulas et al. [13] in 1997. A few years later in 2004, Parsopoulos and Vrahatis [19] employed deflection for the enhancement of the particle swarm optimization algorithm. Specifically, it was used for the detection of all minimizers in various problems, including multimodal optimization test problems,

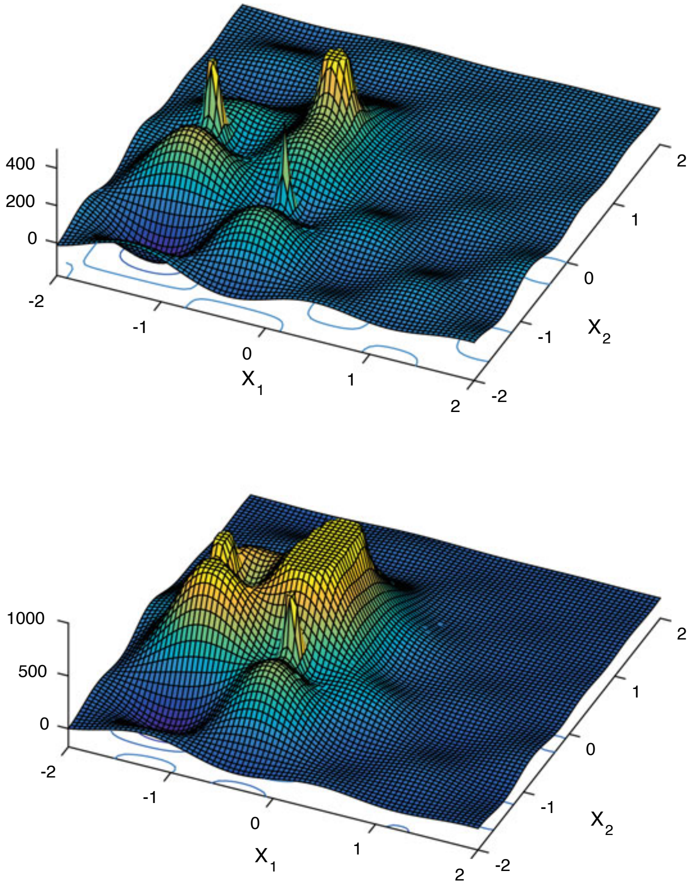

Fig. 4 Deflection transformation concurrently applied on the local minimizers $\mathbf{x}_1 = (-0.3521, -0.8003)^T$, $\mathbf{x}_2 = (-1.3069, -0.1957)^T$, and $\mathbf{x}_3 = (-0.3557, 0.3330)^T$, for $\lambda = 2$ (up) and $\lambda = 1$ (down)

and periodic orbits of nonlinear mappings. The obtained results were impressive for two variants of the studied algorithm.

Motivated by the success of deflection, Pavlidis et al. [21] proposed in 2005 the computation of multiple Nash equilibria in finite strategic games from the field of game theory through computational intelligence methods. Covariance matrix adaptation evolution strategies, differential evolution, and particle swarm optimization equipped with deflection were tested on formal games of 3 up to 5 players with very promising results.

In 2006, Gao and Tong [4] proposed UEAS, a space contraction technique that employed deflection (and stretching as well) in a general iterative optimization scheme that was successfully combined with the differential evolution algorithm.

The experimental setup followed closely that of Parsopoulos and Vrahatis [19], considering similar test problems. In the same year, Zhang and Liu [36] followed the pioneering work in [21]. Specifically, n-players strategic games were considered and deflection was used for learning all Nash equilibria by deflecting the players' payoff function. Experimental results for games with 2 and 3 players revealed the promising behavior of the proposed approach.

Tong et al. extended in 2008 the application range of the UEAS approach to the hardware/software partitioning problem [32]. Deflection was adopted as a crucial part of the search procedure for the detection of multiple partitioning configurations aiming at the minimization of implementation time and cost. In the same year, Zhang and Fan [35] extended the previous work of [36] for learning Nash equilibria through an adaptive policy gradient employing deflection as the mechanism for detecting multiple equilibria.

The ability of deflection in detecting global minimizers was further verified in 2009 by Li et al. [9]. In this case, the chaotic ant swarm was used as the main search algorithm. The dynamics of the algorithm were analyzed on several test problems, showing high ability of detecting all minimizers. In the same year, the previous applications of deflection-equipped algorithms on the detection of Nash equilibria were extended by Liu and Dumitrescu [12]. In their work, a new metaheuristic, namely, roaming optimization, was considered along with differential evolution and particle swarm optimization. Experimental results verified that deflection was beneficial also for the new algorithm.

Chaos control and detection of periodic orbits of chaotic systems were investigated by Gao et al. [3] in 2012. This work followed the basic analysis of the pioneering work in [19] and considered also different chaotic systems. Also, a zooming procedure proved to be beneficial when combined with deflection for the specific application.

Finally, deflection recently constituted an essential mechanism for the production of training samples in a computationally efficient approach based on Gaussian process regression to assess the accessibility in the main-belt asteroids, as proposed in 2017 by Shang and Liu [31]. Numerical simulations demonstrated that the proposed method could achieve significant results with minimal time requirements.

3 Stretching Technique

In the following paragraphs, the basic stretching scheme is presented followed by a number of applications.

3.1 Basic Scheme

The stretching technique was originally introduced by Parsopoulos et al. [15] as a tool for the alleviation of local minima in metaheuristic optimization. It is based on the same essential properties of filled functions as the deflection technique, although its use is recommended in different cases. Specifically, stretching shall be used in environments with large number of local minimizers because it can eliminate in one application a given local minimizer as well as all higher minimizers, while leaving lower minimizers unaffected.

Making our description more concrete, let us assume again the problem of Eq. (1) with the accompanying assumption of a lower bounded objective function $f(\mathbf{x})$. Then, given a detected local minimizer $\mathbf{x}^*$, stretching consists of the following two transformations:

$$G(\mathbf{x}) = f(\mathbf{x}) + \gamma_1 \|\mathbf{x} - \mathbf{x}^*\| \left(\text{sign}\left(f(\mathbf{x}) - f(\mathbf{x}^*)\right) + 1\right), \tag{6}$$

$$H(\mathbf{x}) = G(\mathbf{x}) + \gamma_2 \frac{\text{sign}\,(f(\mathbf{x}) - f(\mathbf{x}^*)) + 1}{\tanh\,(\lambda\,(G(\mathbf{x}) - G(\mathbf{x}^*)))}, \tag{7}$$

where $\gamma_1, \gamma_2 > 0$, are control parameters of the transformations; $\lambda > 0$ is the control parameter of the hyperbolic tangent function; and $\text{sign}(\cdot)$ is the well-known three-valued sign function defined as:

$$\text{sign}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0. \end{cases}$$

The transformation $G(\mathbf{x})$ of Eq. (6) is responsible for "stretching" the objective function upwards, while leaving unaffected all points of lower function values than the detected minimizer. This way, all local minima that are higher than the detected one are eliminated. Then, the transformation $H(\mathbf{x})$ of Eq. (7) transforms the detected minimizer to a maximizer, similarly to the deflection approach. Thus, there is double gain of eliminating not only the detected minimizer but also all minimizers with higher values as well.

A direct consequence of this property is that, contrary to the previously presented deflection technique, stretching shall be applied only on the best local minimizer found so far. This property also renders stretching inappropriate for detecting all minimizers of the objective function because it eliminates multiple local minima in one application. Moreover, if it is applied on a global minimizer it eliminates also all the other global minimizers.

The mexican hat effect is present also in stretching, introducing artificial local minimizers around the targeted one. However, their depth can be controlled through the control parameters of the technique. Let us consider again the Levy no. 5 function of Eq. (5). The transformation $G(\mathbf{x})$ of Eq. (6) applied on the local minimizer $\mathbf{x} = (-0.3572, 1.3166)^T$ is depicted in the upper part of Fig. 5 for $\gamma_1 = 200$. We can clearly observe that the parts of the objective function with values equal or greater than the stretched minimizer are widely affected, while lower minima remain unaffected appearing as holes on the stretched landscape. Also, we can see that the targeted minimizer remains a minimizer after this transformation.

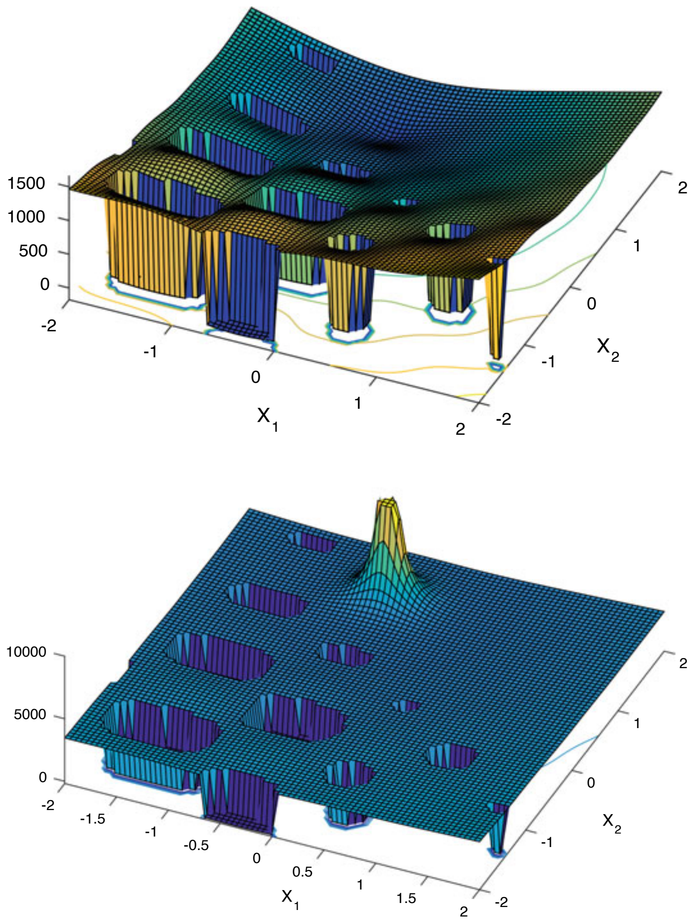

Fig. 5 Stretching transformation $G(x)$ (up) and $H(x)$ (down) applied on the local minimizer $\mathbf{x} = (-0.3572, 1.3166)^T$ for $\gamma_1 = \gamma_2 = 50$ and $\lambda = 0.05$

Retaining $\gamma_1 = 200$, the transformation $H(\mathbf{x})$ of Eq. (7) applied on the same local minimizer is depicted in the lower part of Fig. 5 for $\gamma_2 = 1000$ and $\lambda = 0.005$. Evidently, this transformation turns the targeted minimizer into a maximizer while leaving all the deeper minima unaffected. The parameters γ_2 and λ have a direct effect on the shape of the transformed objective function around the minimizer. Smaller values of γ_2 may produce the mexican hat effect. On the other hand, λ has the opposite effect, namely smaller values raise the maximizer higher, thereby ameliorating the mexican hat effect. Taking into consideration the interesting interplay of the two parameters, an inexperienced user may find it easier to keep the one parameter constant to a reasonable value and tune the other one.

3.2 Variants and Applications

Stretching has been combined with both gradient-based and stochastic optimization algorithms, including trajectory-based and population-based methods. Its flexibility and effectiveness has established it as a viable approach in the relevant literature. So far applications have appeared in software/hardware partitioning, global optimization, semi-infinite programming, red blood cell motion, clustering, and process control problems.

Stretching was initially introduced by Parsopoulos et al. in 2001 [15] for the alleviation of local minima in evolutionary optimization [14, 16, 17], and it was further extended in subsequent works [18, 19]. These works provided a general framework for the application of stretching along with an experimental justification based on a number of standard test problems.

In 2007, Wang and Zhang [33] extended the previous pioneering works by considering stretching with an algorithm that combines gradient search and simulated annealing. Their experiments showed that applying stretching on a local minimizer detected through gradient descent and, subsequently, applying simulated annealing on the stretched function can be highly beneficial.

Later, in 2008, Tong et al. successfully applied a scheme that combined deflection and stretching on the hardware/software partitioning problem in 2008 [32]. In the same year, Wang and Zhang [34] proposed an auxiliary function method that implements the stretching transformation such that it remains descending wherever the function values are higher than the stretched minimum. This was a step toward addressing the mexican hat effect described in the previous sections. This method was later integrated with memetic algorithms in [2].

In 2009, Pereira and Fernandes [26] proposed a global reduction method for solving semi-infinite programming problems. The algorithm was based on simulated annealing, which incorporated a sequence of local applications of stretching to compute the solutions of the lower-level problem. A penalty method was also used for the approximation of the solution of the finite reduced optimization problem, along with a globalization procedure based on line search to guarantee a sufficient decrease of the underlying merit function.

The same authors presented in [22] and later extended in [23] the aforementioned method of simulated annealing with stretching to solve constrained multi-global optimization problems, where all global solutions are sequentially computed by applying stretching with an adaptive simulated annealing variant, while constraint-handling is carried out through a non-differentiable penalty function.

The stretched simulated annealing approach was used also by Pinho et al. [28] for the characterization of the behavior of red blood cell motion through a glass microchannel. Specifically, the radial displacement of forty red blood cells was considered and different functions were used to approximate their displacement by means of global optimization.

Ribeiro et al. [29] proposed the first attempt of applying stretching in a parallel computing environment. Specifically, they considered the problem of solving multilocal programming problems through a parallel stretched simulated annealing, which is based on partitioning the search space. This way, the resolution of the search domains is increased, facilitating the discovery of new solutions while retaining time efficiency. This approach was further analyzed in [24] and extended for the constrained multi-local optimization problem in [25, 30].

In 2013, Lu et al. [11] employed a gradient-based update procedure for the parameters of the kernel-based fuzzy c-means algorithm and tackled the local minima problem by using stretching. Experiments on both artificial and real-world datasets showed that the stretched algorithm with optimized kernel parameters was superior to other competing algorithms.

He et al. [7] proposed in 2014 a combination of stretching with the simulated annealing algorithm, which is iteratively applied on the stretched function, while its trial points generation scheme is especially designed to promote diversity. Numerical results from a large number of test problems suggest that the hybrid method is effective for global optimization.

In 2015, Drąg and Styczeń [1] applied stretching for the control and optimization of multistage technological processes. In their study, the multistage differential-algebraic constraints with unknown consistent initial conditions were considered. The infinite-dimensional optimal control problem was transformed into a finite-dimensional optimization task through the direct shooting method, while simulated annealing with stretching was successfully used to solve the corresponding constrained optimization problem.

4 Experimental Evaluation

Table 1 reports experimental results from the application of deflection with the constriction coefficient version of particle swarm optimization on several test problems, as well as on the problem of detecting periodic orbits of the Hénnon nonlinear map-

Table 1 Numerical results from the application of deflection on various test problems reported in [19]

Problem	Range	Minimizers	Average iterations total/Per minimizer
Egg-holder (TP_{UO-21} in [20])	$[-5, 5]^2$	12/12	315 / 26.2
Levy no. 3	$[-10, 10]^2$	10	2913 / 291.3
Levy no. 5	$[-10, 10]^2$	2	3026.5 / 1036.5
Hénnon mapping ($\cos a = 0.24$)	$[-1, 1]^2$	11/11	249 / 22.6
Hénnon mapping ($\cos a = 0.8$)	$[-1, 1]^2$	2/2	21 / 10.5
Battle of sexes game		3	52 / 17.3

ping and the detection of Nash equilibria for the "battle of sexes" game, as they appear in [19]. For the complete definition of each problem and the exact experimental configuration the reader is referred to [19] and citations therein.

For the case of the egg-holder function and the two versions of the Hénnon mapping, the target was the detection of all global minimizers. Therefore, deflection was applied in combination with the repulsion strategy proposed in [19] as a simple mean to ameliorate the consequences of the mexican hat effect by repelling the particles from the deflected minimizers.

For the rest of the problems, stretching was used to eliminate the multitude of local minima in the ranges of interest. It shall be noted that the two Levy-family problems contain hundreds of local minimizers. Stretching was applied after the detection of the first local minimizer, allowing the algorithm to eventually converge to the global minimizer. Similar performance was achieved for the game theory problems. The reader is referred to [19, 20] for a complete analysis of the results.

5 Conclusions

Since their introduction almost 20 years ago, deflection and stretching have served as the means for locating multiple minimizers and alleviating local minimizers in various applications. During the past years they have been combined with a multitude of algorithms including evolutionary and swarm intelligence methods such as particle swarm optimization, differential evolution, ant colony optimization, and covariance matrix adaptation evolution strategies, as well as trajectory-based stochastic optimization approaches such as simulated annealing, and gradient-based algorithms.

Besides the benchmarking on a plethora of standard test problems, the ongoing scientific activity has resulted in a number of interesting applications from various fields. These include the detection of Nash equilibria from game theory, the detection of periodic orbits and chaos control in nonlinear mappings and chaotic systems, the study of the accessibility in main-belt asteroids from astrophysics, semi-infinite programming, the study of red blood cells motion from medical informatics, the kernel-based fuzzy c-means algorithm from clustering, as well as control and optimization problems.

Deflection and stretching have attracted the interest of the scientific community for many years. It is the authors' belief that the ongoing necessity for multimodal optimization algorithms will add further merit to these approaches, especially in terms of new applications, integration with other algorithms, and theoretical investigations.

Acknowledgements The authors would like to thank the editors for their invitation to contribute in this volume as well as the reviewers for their constructive comments.

References

1. Drąg, P., Styczeń, K.: Simulated annealing with constraints aggregation for control of the multistage processes. In: Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 461–469. IEEE (2015)
2. Fan, L., Wang, Y.: A minimum-elimination-escape memetic algorithm for global optimization: MEEM. Int. J. Innov. Comput. Inf. Control **8**(5(B)), 3689–3703 (2012)
3. Gao, F., Qi, Y., Yin, Q., Xiao, J.: Solving problems in chaos control through an differential evolution algorithm with region zooming. Appl. Mech. Mater. **110–116**, 5048–5056 (2012)
4. Gao, F., Tong, H.: UEAS: a novel united evolutionary algorithm scheme. In: King, et al. (eds.) Lecture Notes in Computer Science (LNCS), vol. 4234, pp. 772–780. Springer (2006)
5. Ge, R.P.: A filled function method for finding a global minimizer of a function of several variables. Math. Program. **46**, 191–204 (1990)
6. Ge, R.P., Qin, Y.F.: A class of filled functions for finding global minimizers of a function of several variables. J. Optim. Theory Appl. **54**(2), 241–252 (1987)
7. He, J., Dai, H., Song, X.: The combination stretching function technique with simulated annealing algorithm for global optimization. Optim. Methods Softw. **29**(3), 629–645 (2014)
8. Horst, R., Tuy, H.: Global Optimization - Deterministic Approaches. Springer, New York, USA (1996)
9. Li, L., Yang, Y., Peng, H.: Computation of multiple global optima through chaotic ant swarm. Chaos, Solitons Fractals **40**, 1399–1407 (2009)
10. Li, X., Epitropakis, M.G., Deb, K., Engelbrecht, A.: Seeking multiple solutions: an updated survey on niching methods and their applications. IEEE Trans. Evol. Comput. **21**(4), 518–538 (2017)
11. Lu, C., Zhu, Z., Gu, X.: Kernel parameter optimization in stretched kernel-based fuzzy clustering. In: Zhou, Z.-H., Schwenker, F. (eds.) Lecture Notes in Artificial Intelligence (LNAI), vol. 8183, pp. 49–57. Springer (2013)
12. Lung, R.I., Dumitrescu, D.: Evolutionary multimodal optimization for Nash equilibria detection. In: Krasnogor, N. et al. (eds.) Studies in Computational Intelligence: Nature Inspired Cooperative Strategies for Optimization, vol. 236, pp. 227–237. Springer (2009)
13. Magoulas, G.D., Vrahatis, M.N., Androulakis, G.S.: On the alleviation of the problem of local minima in back-propagation. Nonlinear Anal. Theory Methods Appl. **30**(7), 4545–4550 (1997)
14. Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Improving particle swarm optimizer by function "stretching". In: Hadjisavvas, N., Pardalos, P.M. (eds.) Advances in Convex Analysis and Global Optimization, vol. 54 of Nonconvex Optimization and Applications, chap. 28, pp. 445–457. Kluwer Academic Publishers, The Netherlands (2001)
15. Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Objective function "stretching" to alleviate convergence to local minima. Nonlinear Anal. Theory, Methods Appl. **47**(5), 3419–3424 (2001)
16. Parsopoulos, K.E., Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Stretching technique for obtaining global minimizers through particle swarm optimization. In: Proceedings of the Particle Swarm Optimization Workshop, pp. 22–29. Indianapolis (IN), USA (2001)
17. Parsopoulos, K.E., Vrahatis, M.N.: Modification of the particle swarm optimizer for locating all the global minima. In: Kurkova, V., Steele, N.C., Neruda, R., Karny, M. (eds.) Artificial Neural Networks and Genetic Algorithms. Computer Science, pp. 324–327. Springer, Wien, Austria (2001)
18. Parsopoulos, K.E., Vrahatis, M.N.: Recent approaches to global optimization problems through particle swarm optimization. Nat. Comput. **1**(2–3), 235–306 (2002)
19. Parsopoulos, K.E., Vrahatis, M.N.: On the computation of all global minimizers through particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 211–224 (2004)
20. Parsopoulos, K.E., Vrahatis, M.N.: Particle Swarm Optimization and Intelligence: Advances and Applications. Information Science Publishing (IGI Global), Hershey (PA), USA (2010)
21. Pavlidis, N.G., Parsopoulos, K.E., Vrahatis, M.N.: Computing Nash equilibria through computational intelligence methods. J. Comput. Appl. Math. **175**, 113–136 (2005)

22. Pereira, A.I., Fernandes, E.M.G.P.: Constrained multi-global optimization using a penalty stretched simulated annealing framework. In: AIP Conference Proceedings 1168, vol. 2, pp. 1354–1357. AIP (2009)
23. Pereira, A.I., Ferreira, O., Pinho, S.P., Fernandes, E.M.G.P.: Multilocal programming and applications. In: Zelinka, I. et al. (eds.) Handbook of Optimization, ISRL 38, pp. 157–186. Springer (2013)
24. Pereira, A.I., Rufino, J.: Solving multilocal optimization problems with a recursive parallel search of the feasible region. In: Murgante, B., et al. (ed.) Lecture Notes in Computer Science (LNCS), vol. 8580, pp. 154–168. Springer (2014)
25. Pereira, A.I., Rufino, J.: Solving constrained multilocal optimization problems with parallel stretched simulated annealing. In: Gervasi, O., et al. (ed.) Lecture Notes in Computer Science (LNCS), vol. 9156, pp. 534–548. Springer (2015)
26. Pereira, A.I.P.N., Fernandes, E.M.G.P.: A reduction method for semi-infinite programming by means of a global stochastic approach. Optimization **58**(6), 713–726 (2009)
27. Petalas, Y.G., Parsopoulos, K.E., Vrahatis, M.N.: Stochastic optimization for detecting periodic orbits of nonlinear mappings. Nonlinear Phenom. Complex Syst. **11**(2), 285–291 (2008)
28. Pinho, D., Pereira, A.I., Lima, R.: Motion of red blood cells in a glass microchannel: a global optimization approach. In: AIP Conference Proceedings 1168, vol. 2, pp. 1362–1365. AIP (2009)
29. Ribeiro, T., Rufino, J., Pereira, A.I.: PSSA: parallel stretched simulated annealing. In: AIP Conference Proceedings 1389, pp. 783–786. AIP (2011)
30. Rufino, J., Pereira, A.I., Pidanic, J.: coPSSA: constrained parallel stretched simulated annealing. In: 25th International Conference Radioelektronika. IEEE (2015)
31. Shang, H., Liu, Y.: Assessing accessibility of main-belt asteroids based on Gaussian process regression. J. Guid. Control Dyn. **40**(5), 1144–1154 (2017)
32. Tong, Q., Zou, X., Tong, H., Gao, F., Zhang, Q.: Hardware/software partitioning in embedded system based on novel united evolutionary algorithm scheme. In: Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE 2008), Phuket, Thailand. IEEE (2008)
33. Wang, Y.-J., Zhang, J.-S.: An efficient algorithm for large scale global optimization of continuous functions. J. Comput. Appl. Math. **206**, 1015–1026 (2007)
34. Wang, Y.-J., Zhang, J.-S.: A new constructing auxiliary function method for global optimization. Math. Comput. Model. **47**, 1396–1410 (2008)
35. Zhang, H., Fan, Y.: An adaptive policy gradient in learning Nash equilibria. Neurocomputing **72**, 533–538 (2008)
36. Zhang, H., Liu, P.: A momentum-based approach to learning Nash equilibria. In: Shi, Z., Sadananda, R. (eds.) Lecture Notes in Artificial Intelligence (LNAI), vol. 4088, pp. 528–533. Springer (2006)