Annals of Mathematics and Artificial Intelligence (2020) 88:3–37
https://doi.org/10.1007/s10472-019-09661-7

Dynamic search trajectory methods for global optimization

Stamatios-Aggelos N. Alexandropoulos[1] · Panos M. Pardalos[2] · Michael N. Vrahatis[1]

Published online: 27 August 2019

Abstract
A detailed review of the dynamic search trajectory methods for global optimization is given. In addition, a family of dynamic search trajectories methods that are created using numerical methods for solving autonomous ordinary differential equations is presented. Furthermore, a strategy for developing globally convergent methods that is applicable to the proposed family of methods is given and the corresponding theorem is proved. Finally, theoretical results for obtaining nonmonotone convergent methods that exploit the accumulated information with regard to the most recent values of the objective function are given.

Keywords Dynamic search trajectories · Trajectory methods · Autonomous initial value problems · Globally convergent algorithms · Nonmonotone convergent strategies · Global optimization · Neural networks training

Mathematics Subject Classification (2010) 65K05 · 65K10 · 65L05 · 68T05 · 68T20

1 Introduction

The problem of the mathematical optimization is one of the most known and widely studied problems, not only by mathematicians but by many other scientists all over the world. This is so because the applications that require the solution of an optimization problem are many and varied. In addition, several important optimization problems remain open [49].

✉ Michael N. Vrahatis
vrahatis@math.upatras.gr

Stamatios-Aggelos N. Alexandropoulos
alekst@math.upatras.gr

Panos M. Pardalos
pardalos@ufl.edu

[1] Computational Intelligence Laboratory – CILab, Department of Mathematics, University of Patras, GR-26110 Patras, Greece

[2] Center for Applied Optimization – CAO, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

Indicatively, we mention a few scientific fields that constantly encounter such optimization problems including industry, robotics, mathematics, physics, chemical engineering, economics, machine learning, computational biology, geology, engineering, among others. Although the scientific community has studied the problem of optimizing objective functions for many years, this problem remains attractive [3, 14, 18, 48, 49, 77, 78, 92].

We shall allow us to briefly discuss a few basic concepts regarding the problem of global optimization [33]. The global optimization problem concerns in the computation of the global minimum or maximum of an objective function defined on a given set. Usually the global optimization problem is referred as a global minimization problem. Assume that $F : \mathscr{D} \subset \mathbb{R}^n \to \mathbb{R}$ is a nonlinear and non-convex continuous function and $\mathscr{D}$ is a compact set not necessarily convex. Suppose that $F^\star$ is the global minimum of the function F and $X^\star = \left\{x^\star \in \mathscr{D} \;\middle|\; F(x^\star) = F^\star\right\}$ is the set of all the global minimizers of F in $\mathscr{D}$. Then the global minimization problem is given as $\min_{x\in\mathscr{D}} F(x)$, that is to find a global minimizer $x^\star \in X^\star$. It is well known that in general, global optimization is a difficult to tackle problem. In particular, it is known that a deterministic method for global optimization is NP-hard [85]. On the other hand, there is evidence that a stochastic algorithm can be executed in polynomial time, on the average. Also, theoretical analysis of random search methods indicates that performance may be very good, possibly polynomial in dimension [93]. Furthermore, due to "No free lunch theorems for optimization" [89] (for a recent review see [1]) there is not a single algorithm that performs well on all problems and if an algorithm is improved for one particular problem, it will not perform well for other problems.

In order to tackle the difficulty of the global optimization problem stochastic adaptive search methods for global optimization can be used. Various such methods are nature inspired and many of them are called *heuristic methods* or *population based* methods [79]. This family include the well-known and widely used *differential evolution* (DE) algorithms [80], the *particle swarm optimization* (PSO) algorithm [50, 52, 67], the *evolutionary algorithms* (EA) [4] among others. It is worth mentioning that quite often encountered in the literature hybrid techniques that utilize algorithms of the above category and methods that exhibit local convergence [23] in order to exploit the advantages of both categories local and global convergence. In addition, in many problems [46, 51], two or more objective functions need to be minimized at the same time. Moreover, a problem that requires single-objective optimization can be transformed into another one, for which two or more functions have to be optimized simultaneously. This problem is called *multi-objective* or *multiple criteria* optimization problem and we refer the interested reader to [95].

An effective solution to the problem of global optimization can be obtained using ordinary differential equations solvers. In particular, approaches that are based on the dynamic search trajectories [11, 29, 69, 71] have been used and have provided very good results. In detail, several methods have been developed over the last thirty years that adopt the following strategy:

> Given a second-order (or first-order) differential equation and a function F, we are looking for the minimum value of F using a specific differential equation. In particular, assuming that the differential equation defines the motion of a particle and the function F the potential energy of the particle, then the solution of this equation is called trajectory and determines a possible solution of F. Given some initial conditions and properties (which are analytically presented in the main body of this paper) the particle follows the trajectory to the minimum value of the function under consideration and, thus, the initial problem is solved.

Following the above procedure, one may expect that at the end we will reach the global minimum of the function. This, however, cannot be true since in various cases take place the exact opposite. To this end, methods utilizing dynamic search trajectories are ideally combined with meta-heuristic or other multi-start techniques in order to ensure global optimization as well as the global convergence of the scheme that is used. The trajectories that have been used are varied and their applications diverse between different scientific fields.

In this paper we aim to perform a detailed review of optimization methods that use dynamic search trajectories. The description deals with both past and recent approaches. In addition, we present a new class of dynamic search trajectories methods that are created using numerical methods for solving autonomous ordinary differential equations. Furthermore, we give a strategy for developing globally convergent methods that is applicable to the proposed class of methods as well as we give a proof and the corresponding theorem. Finally, we give theoretical results for obtaining "nonmonotone" convergent methods that exploit the accumulated information with regard to the most recent values of the objective function.

The rest of the paper is organized as follows. In the next section, we present a survey of well-known and widely used trajectory methods that have been studied in the literature, such as the well-known dynamic search trajectory Snyman-Fatti method [71]. In Section 3, we provide the most recent dynamic search trajectory methods and their applications, focusing on up-to-date methods developed over the last ten years. In addition, in the same section we provide a table with the most cited papers, in order to inform the reader about the applicability and the interest gathered regarding trajectory methods. In Section 4 we present a new class of trajectory methods for global optimization based on the numerical solution of autonomous ordinary differential equations. Furthermore, we present a strategy for developing globally convergent methods as well as we give a proof and the corresponding theorem. Furthermore, we give theoretical results for obtaining "nonmonotone" convergent methods. Finally, the paper ends with a synopsis and a short discussion for future research work.

2 Survey of well-known and widely used trajectory methods and their applications

In this section, we provide the first attempts that use dynamic search trajectories in order to create reliable optimization methods. Thus, for every method we present the basic equations, the basic notations and the main philosophy behind the presented approach.

2.1 Dynamic search trajectory methods

One of the first attempts in which the reader meets the combination of differential equations with numerical techniques was made by Branin [11]. In this paper, the author tackles the problem of finding the solution of a system of non-linear equations. In particular, the proposed method can identify multiple roots and, in some cases, achieves global convergence. In addition, if a transformation occurs, his method can locate, under certain conditions, many extrema points of a continuous function. The provided scheme has been tested on a well-known electrical problem, called *tunnel diode* problem. The differential equation that used at the beginning of his attempt was the following:

$$\frac{dx}{dt} + f(x) = 0, \tag{1}$$

where $f(x) = 0$ represents the system of non-linear equations that has to be solved. However, the above equation was replaced by the following first-order differential equation:

$$\frac{df}{dt} + f(x) = 0, \tag{2}$$

since (1) was found inappropriate regarding other problems beyond *tunnel diode* problem. Concerning the analytical solution:

$$f(x(t)) = f(x(0))\, e^{-t}, \tag{3}$$

it is quite accepted that $f(x) = 0$, as $t \to \infty$. Applying the chain rule and multiplying from the left both sides of (3) with the inverse of the Jacobian matrix, we obtain the following equation:

$$\begin{aligned} \tfrac{dx}{dt} &= -\left(\tfrac{\partial f}{\partial x}\right)^{-1} f(x) \\ &= -J_{f(x)}^{-1}\, f(x). \end{aligned} \tag{4}$$

By applying the well-known Euler method, we get the following iterative scheme:

$$x_{n+1} = x_n \mp h_n\, J_{f(x_n)}^{-1}\, f(x_n), \qquad n = 0, 1, 2, \ldots \tag{5}$$

where h_n is the step size at the n-th iteration. This method is known as *Newton trajectory method*. It is worth noting that the sequence of the trajectories given by this method makes sense only in the case where the Jacobian matrix of the objective function is non-singular. In the case where the Jacobian matrix is singular, Branin in [11] suggested to allow a change of sign in (2). Thus, the corresponding equation is written as follows:

$$\frac{df}{dt} \pm f(x) = 0, \tag{6}$$

with solution:

$$f\big(x(t)\big) = f\big(x(0)\big)\, e^{\mp t}. \tag{7}$$

Therefore, when the determinant of the Jacobian matrix changes sign (after passing through zero), the sign in (6) is reversed and the integration procedure may be continued. In this case we obtain:

$$\frac{dx}{dt} = \mp J_{f(x)}^{-1}\, f(x). \tag{8}$$

Branin tested the proposed method to the problems: (a) *two– dimensional* and *three–dimensional trigonometric* problem, (b) *Brent's* problem and (c) *Rosenbrock's* problem. He pointed out that the produced trajectory path, in many cases, passes through all the solutions that he was looking for.

The authors in [35] proposed a iterative process for solving nonlinear equations. The proposed method associates systems of ordinary differential equations with the equations of a non-linear system. Different equations have been used. They are inspired by classical mechanisms of nature and are, mainly, second order ordinary differential equations. An advantage of the provided method is that it can easily handle problems of nonlinear least squares. The issue of successful convergence and, secondly, the expansion of convergence regions are very important. In the literature [11, 94], several methods have been presented that rely on systems of differential equation in order to solve non-linear algebraic equations. The authors in [35] used a system of second-order differential equations in order to solve a

nonlinear system of equations $F(x) = 0$ where $F = (f_1, f_2, \ldots, f_n) : \mathbb{R}^n \to \mathbb{R}^n$. They considered the function $G : \mathbb{R}^n \to \mathbb{R}$:

$$G(x) = \sum_{i=1}^{n} f_i^2(x), \tag{9}$$

and the following second-order differential equation in order to find the roots of function G:

$$\mu(t)\,\frac{d^2x(t)}{dt^2} + g(t)\left[a(t)I + \big(1 - a(t)\big)\,J(x(t))^\top J(x(t))\right]\frac{dx(t)}{dt} = -\nabla G(x(t)), \tag{10}$$

where $\mu(t)$ and $g(t)$ are real-valued positive functions, $0 < a(t) \leqslant 1, \forall\, t \geqslant 0$, I is the identity matrix and $J(x)$ the Jacobian matrix. The above second-order differential equation represent the Newton's law.

Next, we present a very important theorem as it is formulated in the paper [35], which states that the solutions of the differential equation, under certain conditions, lead to the solutions $x^\star$ of the function G.

Theorem 1 *Assume that $G(x) \in \mathbf{C}^2(\mathbb{R}^n)$ (i.e. $G(x)$ is a twice continuously differentiable function) and the following assumptions are hold:*

1. $\mu(t) \in \mathbf{C}^1([0, +\infty))$, $\dfrac{d\mu}{dt} \leqslant 0$, $\mu(t) \geqslant \mu_{\min} > 0$ *for constant value* $\mu_{\min}$
2. $\exists\, b > 0$ *such that* $\dfrac{g(t)}{\mu(t)}\left\langle z, \Big(a(t)I + \big(1 - a(t)\big)J(x)^\top J(x)\Big)\, z\right\rangle \geqslant \dfrac{b}{2}\langle z, z\rangle, \quad \forall\, x, z \in \mathbb{R}^n$ *and* $t \in [0, +\infty)$, *where the equality in the above formula applies if and only if* $z = 0$.
3. $x^\star$ *is an isolated minimum of* G.

Then $x(t) \equiv x^\star$ *and* $v(t) \equiv 0$ *is a solution of:*

$$\mu(t)\,\frac{dv(t)}{dt} + g(t)\left(a(t)\,I + \big(1 - a(t)\big)\,J(x(t))^\top J(x(t))\right)\dot{v}(t) = -\nabla G(x(t)), \tag{11}$$

where $\dot{v}(t) = \dfrac{dx(t)}{dt}$; *moreover it is uniformly asymptotically stable.*

One can easily conclude that the problem of solving $F(x) = 0$ is equivalent to the minimization of the function $G(x)$. Furthermore, this theorem highlights the connection between differential equations, minimization problem, solving a system of nonlinear equations and solving a nonlinear least squares problem (we refer the interested reader to [65]). The solutions of the ordinary differential equation that concern the conditions of Cauchy:

$$\begin{aligned} x(0) &= x^0, \\ \frac{d\,x(0)}{dt} &= v^0, \end{aligned} \tag{12}$$

obtained using a modification of the well-known Euler method. The experiments conducted by the authors have shown that the proposed method is advantageous against Newton-Raphson which has been used in Branin's work [11]. Moreover, the proposed method was tested on five problems, using the *Rosenbrock's* function and the *Boggs's* system of equations [10].

In the paper [27] Griewank has presented some basic properties of the trajectory that need to be fulfilled. Below, we discuss these issues that are related to the function and the trajectory under consideration.

1. First of all, we assume that the given function can be expressed as a sum of two factors:
$$F = g + \varepsilon, \tag{13}$$
where g is a function with a unique minimum, $g^\star = g(y^\star)$ and let ε be a bounded perturbation.
2. We assume that the factor ε is a small amount in relation to the amount of the value of g.
3. We can reach the global minima of F, let $F^\star = F(x^\star)$ and all the other local minima of F is closed to $y^\star$, in other words they belong to the neighborhood of $y^\star$.

By taking into account all the above cases, every technique that is used to reach the global minima of F is possible to reach many local minima outside the neighborhood of $y^\star$, that concern us.

Next, we give the following desirable properties of a search trajectory.

1. The trajectory cannot converge to minima with values greater than a *target level* η which is ideally slightly larger than the global minimum $F(x^\star)$.
2. As long as the value of F is much larger than the target η, the trajectory is not greatly affected by the perturbation ε. As a result, the trajectory needs to follow a descent direction with respect to the negative value of gradient of g, $-\nabla g$.
3. As the value of F approaches to the target η, the trajectory minimizes sufficiently and finally reduces to a local optimization technique when $F \leqslant \eta$.
4. The trajectory does not depend entirely on the Hessian matrix of F, $\nabla^2 F$.
5. Finally, the trajectory is not affected by a transformation into the variables defined above and the multiplication of the quantity $F - \eta$ by any positive scalar element.

Therefore, Griewank used the following second-order differential equation, the solution of which produces a trajectory that meets the above conditions. Thus, in his work [27] verified the above conditions for the equation:

$$\ddot{x}(t) = -\omega\big(I - \dot{x}(t)\,\dot{x}(t)^\top\big)\,\nabla F(x(t))/(F(x(t)) - \eta), \tag{14}$$

where ω is a positive real number. By omitting the variable t, (14) is written as follows:

$$\ddot{x} = -\omega\big(I - \dot{x}\dot{x}^\top\big)\,\nabla F(x)/\big(F(x) - \eta\big), \qquad \omega > 0, \tag{15}$$

for any initial point value $(x_0, \dot{x}_0)$, $F_0 = F(x_0) > \eta$ and $\|\dot{x}_0\| = 1$. The above equation arose from a variant of an earlier work of the author in [26] as a basic precondition for the solution of the following problem:

$$\min_{x(t)\in\mathscr{X}} \int_{t_0}^{t_1} \big(F(x(\tau)) - \eta\big)^{-\omega} d\tau, \tag{16}$$

where the $\mathscr{X}$ is a continuous rectifiable set of paths between two point $x(t_0) \equiv x_0$, $x(t_1) \equiv x_1$ and $\|\dot{x}\| = 1$. In [27] the author thoroughly verifies the validity of the above hypotheses 1–5. In this work, however, he did not perform the analytical verification of these conditions.

Various trajectories have been derived from well-known formulas of Mechanics and Physics [35, 36, 94], such as the equation:

$$m(t)\,\ddot{x}(t) + v(t)\,\dot{x}(t) = -\nabla F(x(t)), \tag{17}$$

which is one of the most known differential equations (Newton's law), for a particle of mass $m(t)$, in a consideration objective F, into a dissipative force $-v(t)\,\dot{x}(t)$. Thus, it is easy to observe that if we set:

$$m(t) = (F(x(t)) - \eta)/\omega, \tag{18}$$

and

$$v(t) = -\nabla F\big(x(t)\big)^{\top}\,\dot{x}(t), \tag{19}$$

and replace them in Relation (5) then (2) is obtained.

At this point, it is worth mentioning that the following problem may arise. If the parameters $m(t)$ and $v(t)$ are positive and some conditions for F are hold, then any trajectory may converge to a local minimum of F. The following formula that has been given by Griewank [27] indicates that there is a tendency that slows the trajectory when uphill conditions are met, such as $\nabla F(x)^{\top}\dot{x} > 0$. The formula under consideration was:

$$\frac{d}{dt}\left[1/2\,\|\dot{x}(t)\|^2\right] = -\frac{1}{m(t)}\left(\nabla F(x(t))^{\top}\,\dot{x}(t) + v(t)\,\|\dot{x}(t)\|^2\right). \tag{20}$$

On the other hand, as the trajectory moves down, we want to accelerate the path to the minimum, reducing the influence of the local gradient. However, it is possible the entrapment of the trajectory to a local minimum. Thus, the trajectory is set in a "dent" on a hill, instead of falling towards a "slope". Therefore, the desired move to the global minimum of the function is not performed.

In [27] the author formulates and proves a theorem (Theorem 3.1), which reinforces the opinion that the solution of the differential equation (2) is transformed into the trajectory of the steepest descent. This happens when the value of F approaches the value of η. This allows the trajectory to be minimized as much as is needed and ultimately, creates the following extension: *Any finite solution of (2) may resulted by the formula:*

$$\dot{x}(t) = -\nabla F(x)/\|\nabla F(x)\|. \tag{21}$$

The resulting search trajectories are uniquely determined by the initial conditions $(x_0,\ \dot{x}_0)$ and solve the system:

$$\max\{0,\ F(x) - \eta\}\,\ddot{x} + \omega\big(I - \dot{x}\,\dot{x}^{\top}\big)\,\nabla F(x) = 0. \tag{22}$$

Finally, if the solutions do not approach η, the solutions of the above system converge either to a minimum of F or to a possible saddle point of F. In particular, in his work he tried to calculate the minima of an objective function with many local minima at low cost. In order to achieve this goal, the author considered the trajectories produced by the differential equation (2), in relation to the parameters η and ω. One may expect that the overall trajectories would be acceptable. However, this could not be true. Thus, only the trajectories that are close to the target values η are "alive". On the contrary, trajectories' values which are much higher than the value of η are rejected. The author considered that it is possible to make this (to produce "good" trajectories), as long as the relation between the ω parameter and the dimension of the function under consideration is correctly determined. Furthermore, he gave an analysis of the value of ω.

The problem that we have described above, the trajectories can "wander" in neighborhoods where reside unacceptable minima or even to exceed the acceptable ones, is extremely important. For this reason, the author tackled this issue by constructing a procedure, which uses the solutions of the (2) in order to conduct global search. Thus, by setting successively low values for the target parameter η, in combination with conventional local minimization techniques, he overcame the above-mentioned difficulty. After suitable parametrization

and transformation of the (2), the author designed an one step method of convergence order three. This new method tested on well-known functions, such as (a) *six-hump function*, (b) *camel-back function* and (c) a *quadratic form function*. The results have shown that Griewank's method is quite competitive in order to compute local minima. The computational cost was high (function and gradient evaluations) but it is comparable to the proposed techniques so far.

In the work of Snyman [69] a practical localization method of local minima has been proposed, mainly for functions for which the first derivative can be easily estimated. This method has been aimed for locating the minima of multivariable functions. On the one hand, the proposed method was a naive approach and several improvements may be done. On the other hand, it has a number of advantages, which are worth to be discussed. Snyman's method compared with Fletcher's program and the well-known quasi-Newton method. The results have shown that the method is reliable and robust on a set of standard test functions. In particular, it approached the required local minimum in all the test cases, in contrast, for example, to the quasi-Newton method that in some cases simply approached close to the minima. As indicated above, the method is conceptually very simple, and the Fortran coding program is fairly easy to understand and easy to be used. In addition, the storage requirements, which were extremely important for that time, were minimal. Specifically, the method required the storage of five n-dimensional vectors, unlike to the quasi-Newton which required the storage of the approximations of the Hessian matrix.

Considering the overall performance of this method, one would say that it performs particularly well on functions that present steep narrow "valleys". Moreover, Snyman's method responds very well in cases where the starting point is not close to the local minima. On the contrast, in cases where the objective function presents flat minima, its performance was not as good as the described case. Nevertheless, the method remains competitive and not so less reliable related to the quasi-Newton. In his original work [69] the author proves that the computational time required by his method in order to achieve convergence is increased approximately linearly. In this paper, however, we will not deal with this proof in detail. However, it is worth mentioning that for the performed tests in [69] the gradient evaluations of the corresponding functions were quite inexpensive. This obviously favors the dynamic method presented by Snyman. The test functions that have been used were: (a) the *parabolic valley function* ($n = 2, 4, 24$), (b) the *cubic valley function*, (c) the *Beale's function*, (d) the *Powell's function*, (e) the *Wood's function*, (f) the *homogeneous quadratic function* ($n = 40$) and (g) the *Oren's power function* ($n = 20$). Snyman's dynamic method was compared with:

(a) Fletcher's program, called "VAO9A", which uses the well-known quasi-Newton scheme [22, 25]
(b) Davidon – Fletcher – Powell (DFP) method (code of [34]).

The optimization problem under consideration is defined below:

$$\min F(x), \qquad x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n, \qquad F \in \mathbf{C}^1. \tag{23}$$

Assume that $F(x)$ represents the potential energy of a particle of unit mass in a n-dimensional force field, then, if the function F has a local minimum at $x^\star$, it follows that:

$$F(x) = -\int_{x^\star}^{x} a(s)^\top ds + F(x^\star), \tag{24}$$

where $a(s)$ represents a force acting on the particle at the point s. The kinetic energy of the particle is defined by:

$$T(x) = \frac{1}{2}\sum_{i=1}^{n}\dot{x}_i^2 = \frac{1}{2}\|\dot{x}\|^2, \tag{25}$$

and the Lagrangian is given by:

$$L(x) = T(x) - F(x) = \frac{1}{2}\sum_{i=1}^{n}\dot{x}_i^2 + \int_{x^\star}^{x} a(s)^\top ds - F(x^\star). \tag{26}$$

By applying Hamilton's principle the following equations of motion are obtained:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_i}\right) - \frac{\partial L}{\partial x} = 0, \quad i = 1, 2, \ldots n. \tag{27}$$

By making the appropriate replacements we obtain the following differential equation:

$$\ddot{x} = -\nabla F = a. \tag{28}$$

The trajectory $x(t)$ of the particle is given by the solution of the (28). Summarizing, we get the problem (28) with initial conditions:

$$\begin{cases} x(0) = x_0, \\ \dot{x}(0) = v(0) = v_0 = 0, \end{cases} \tag{29}$$

where v_0 determines the initial velocity.

Equation (28) implies conservation of energy. Thus, $\forall\, x$, along the trajectory we obtain:

$$T(x) + F(x) = T(x_0) + F(x_0) = E_0, \tag{30}$$

where E_0 determines the initial total energy. Thus, for the function F we get:

$$F(x) = T(x_0) + F(x_0) - T(x) = F(x_0) - T(x). \tag{31}$$

Therefore, with $T(x_0) = 0$ along the particle path we obtain $F(x) \leqslant F(x_0)$. However, given the fact that frictional forces are absent, the following problem may occur: the particle will be in continual motion and therefore there is a risk to overcome the minimizer $x^\star$ of the function. To sum up, it is necessary to ensure that the energy of the particle is reduced. As a result, the trajectory x is following a path to the local minimizer $x^\star$. To achieve this goal, there are two known techniques:

1. The first one uses an artificial damping term:

$$\ddot{x} = -\nabla F - a\dot{x}, \tag{32}$$

   where $a > 0$ is a given damping constant. The successfulness of this strategy depends on the appropriate selection of the parameter a. Thus, we have to tackle a new additional problem: *Which is the optimal value for the term a? Is the selection of this parameter familiar to the position of the trajectory or the nature of the force field?* One may easily understand that this is not a simple problem and the solution is not trivial.
2. The second strategy which is adopted in [69] is based on the monitoring of the kinetic energy of the particle at fixed time intervals. Assume that x_k, v_k are the position and

the velocity of the particle at the time $t_k = k\Delta t$. For any time points t_k and t_{k+1} by combining (30) we obtain:

$$\begin{aligned}\Delta F_k &= F_{k+1} - F_k = F(x_{k+1}) - F(x_k)\\ &= T(x_0) + F(x_0) - T(x_{k+1}) - (T(x_0) + F(x_0) - T(x_k))\\ &= T(x_0) + F(x_0) - T(x_{k+1}) - T(x_0) - F(x_0) + T(x_k)\\ &= -T(x_{k+1}) + T(x_k) = -T_{k+1} + T_k = T_k - T_{k+1}\\ &= -\Delta T_k = -\Delta T(x_k). \end{aligned} \tag{33}$$

Obviously, for $\Delta T_k > 0$ we obtain $\Delta F_k < 0$ that is required.

Snyman in [70] proposed an improvement in order to increase the performance of his original algorithm [69]. This improvement mainly concerned the selection of the parameter *time step*. Particularly, in the provided algorithm, this step is reduced or increased appropriately in an automatic way. This results in a more efficient implementation of the basic dynamic algorithm. This change made a more sophisticated algorithm with much better performance than the original *leap-frog* algorithm which used in the paper [69]. In this paper we do not provide a thorough analysis of these algorithms. For this reason, we suggest studying together these two works of Snyman in order to make the understanding easier. In summary, we present the main difference in the time step parameter of the algorithm.

It is well-known that if the *time step* is very big, then the trajectory may be inaccurate. On the other hand, the trajectory may not be significantly out of reach of the solution, but the suitable situation is to be as close as possible to $x^\star$ in order to achieve the convergence of the algorithm. An useful idea, in order to test if the *time step* is large, is to check the difference between two successive gradient vectors. If these vectors are vertical or form an angle of 90 degrees or greater, then the step is very large. However, one step is not considered as inappropriate at the first observation of such an angle. Thus, at each step we evaluate the quantity $a_k^\top a_{k-1}$, where a_k denotes the negative gradient value of F at the point x_k. Similarly for the amount of a_{k-1}. If this quantity is less than or equal to zero for consecutive steps, then the *time step* must be halved. Then the procedure restarts from the point $(x_k + x_{k+1})/2$ with velocity value $(v_k + v_{k+1})/4$. Typically, the number of successive steps selected is $m = 3$. On the other hand, the following strategy is adopted if the time step is needed to be increased. In order to increase the time stem the following two inequalities must to be hold: $a_{k+1}^\top a_k > 0$ and $\|\Delta x_k\| < \delta$, where δ represents the maximum permissible step size. Therefore, the step changes according to the relationship:

$$p = 1 + N\delta, \qquad \delta > 0, \tag{34}$$

where N represents the successive successful steps that have been carried out. Otherwise, N is reset to the value 1 and is not stepped up.

One of the most known methods of this class is the method proposed by Snyman and Fatti [71]. This method has attracted strong interest by many researchers. For this reason many variations of this method have been provided, which we will briefly present in this review. Next we present the main points of the original method. The authors produced a global minimization method using a multi-start algorithm. The success of this method was that with appropriate modifications to the searching trajectories, the authors managed to broaden convergence regions regarding the global minimum of the objective function. The second-order differential equation that has been used in their approach is the following:

$$\ddot{x}(t) = -\nabla F(x(t)), \tag{35}$$

where the solution x represents the movement of a particle of unit mass inside a conservative force field, F represents the potential energy of this particle and $x(0) = x_0$, $\dot{x}(0) = 0$ the initial conditions. Multiplying the above equation by $\dot{x}(t)$ we obtain the formula of the kinetic energy:

$$\frac{1}{2}\left\|\dot{x}(t)\right\|^2 + F(x(t)) = F(x_0), \tag{36}$$

for $t = 0$. Furthermore, if the following relation is satisfied:

$$-\nabla F^{\top}\dot{x} > 0, \tag{37}$$

then, the kinetic energy of the particle is increased, which means that the function value is decreased through the steepest descent path. As we have already mentioned, it is possible the function under consideration to have many local minima, which, in fact, can be located in their entirety by the trajectories. In order to achieve the goal of identifying the global minimum, it is beneficial to record the current minimum values x_n, $\dot{x}_n$ and F_n. The trajectories continue their path, expecting towards the global minimum. As it is emphasized by the authors [71], it is important to be able to ensure the convergence of the proposed method, at least, to a local minimum. The difficult question, however, is *how we can use the gained by the trajectory information since the current minimum value?* In their work, Snyman and Fatti followed the following approach. Along with the original trajectory, they produced additional trajectories, named as "auxiliary trajectories". Using this approach, we are able to consider that x_n is the best value for which the first trajectory terminates (this one with the smallest function value) and x_0 the initial condition. Then, let us denote the auxiliary trajectory, as x_a as follows:

$$x_a = x_0^2 = \frac{1}{2}(x_0^1 + x_n^1), \tag{38}$$

where x_0^2 represents the auxiliary initial condition. In addition, its velocity type is given by:

$$\dot{x}_a = \dot{x}_0^2 = \frac{1}{2}\dot{x}_n^1. \tag{39}$$

To sum up, we assume that for the initial condition x_0^1, the first trajectory, let T_1, is produced and the corresponding velocity is $\dot{x}_0^1$. Then we record the corresponding values x_n^1, $\dot{x}_n^1$ and F_n^1, which are considered as the best values that T_1 returned. In order to produce the second trajectory, let T_2, we set:

$$\begin{aligned} x_0^2 &= \frac{1}{2}(x_0^1 + x_n^1), \quad \text{and} \\ \dot{x}_0^2 &= \frac{1}{2}\dot{x}_n^1, \end{aligned} \tag{40}$$

and again we record the best values, as x_n^2, $\dot{x}_n^2$ and F_n^2. If the current best function value is smaller than the previous one, i.e. $F(x_n^2) < F(x_n^1)$, then we keep the value of x_n^2, otherwise we set $x_n^2 = x_n^1$ and the same procedure is repeated in order to produce a new auxiliary T_i. The authors presented the following convergence theorem for the described process:

Theorem 2 *Assume that F is the objective function under consideration and $S = \{x : F(x) \leqslant F(x_0^1)\}$ its bounded level set. If the gradient ∇F of F and the sufficient n-order derivatives are exist and are continuous over the set S, then:*

$$\nabla F(x_n^i) = 0, \quad \textit{or} \quad \nabla F(x_n^i) \to 0. \tag{41}$$

Remark 1 It is obvious that a trajectory must be terminated before it begins to retrace.

As a follow-up of a previous study [69], Smyman and Fatti, have used the well-known *leap-frog* integration method in order to provide a reliable discretized implementation of the *auxiliary trajectory* procedure. Next, we present the main points of this approach. The *leap-frog* integration method is described as follows:

$$\begin{aligned} x_{n+1} &= x_n + \dot{x}_n \, \Delta t, \quad n = 0, 1, 2, \ldots \\ \dot{x}_{n+1} &= \dot{x}_n - \nabla F(x_{n+1}) \, \Delta t, \quad n = 0, 1, 2, \ldots \end{aligned} \tag{42}$$

where x_0, $\dot{x}_0$ are the initial conditions, Δt. As it is has been shown in detail in [69], this method suggests energy conservation relationship. Moreover, the initial condition $\dot{x}_0$ is given by:

$$\dot{x}_0 = -\frac{\nabla F(x_0) \Delta t}{\lambda}, \tag{43}$$

where the most common choice of λ is the value $\lambda = 2$. Regarding the time step Δt, it is selected in order to ensure descent at the first step. If this does not happen, then a choice of a different time step value is required. Usually, the time step is divided and the trajectory is restarted until the descent is reached. The method terminates when the following relationship holds:

$$F\big(x(t)\big) - F_n > \kappa \big(F(x_0) - F_n\big), \tag{44}$$

where $x(t)$ determines the trajectory, F_n is the current minimum value of F and κ is commonly valued as $\kappa < 1$ and $\kappa \cong 1$. In [71] the parameter κ is indicated as $\kappa = 0.95$. In addition, if the following relationship is fulfilled:

$$\frac{1}{2} \big\| \dot{x}(t) \big\|^2 < (1 - \kappa) \big(F(x_0) - F_n\big), \tag{45}$$

then, the trajectory is terminated in an uphill path. Furthermore, the minimization process is completed, if for any $x(t)$ the following relationship is satisfied: $\big\| \nabla F(x_n(t)) \big\| < \varepsilon$,

where ε is a small positive value. In order to describe this method we have mentioned that the global goal of the algorithm includes a stochastic term which is inspired in [41]. In particular, it concerns with the probability of the last obtained trajectory that is desirable in order to attain the global minimum. Our presentation about Snyman-Fattis' method ends up, by providing the corresponding theorem [71].

Theorem 3 *Let k be the number of sample points falling within the region of convergence of the current overall minimum, Assume that $\bar{F}$ have been sampled after n points. Then, under the following assumption:*

$$p^\star = Pr[R^\star] = \max_i p_i, \tag{46}$$

and a statistically noninformative prior distribution, the desired probability that $\bar{F}$ be equal to $F^\star$ satisfies the following relationship:

$$Pr[\bar{F} = F^\star] \geqslant q(n, k) = 1 - \frac{(n+1)!\,(2n-k)!}{(2n+1)!\,(n-k)!}, \tag{47}$$

where $R^\star$ represents the region of convergence of the global minimum $F^\star$, p_i represents the probability of a randomly selected point falling into R_i (the region of convergence of a local minimum), while $p^\star$ represents the corresponding probability for the global minimum.

The proof of the theorem is presented in detail in [71]. Regarding the experimental results conducted by the authors, it is worth mentioning that were used nine well-known functions

[17, 27, 62]. The remarkable point was that the proposed *Snyman-Fatti method* successfully returned the global minimum for all the test functions considered in their study.

2.2 Applications of the dynamic search trajectory methods

The differential equations have a wide range of applications in science and technology including aerospace, robotics, vehicles among others. In particular, the introduction of laminates into the above-mentioned fields has led to the need of optimal design of such surfaces in order to optimize the structures. In the paper [40], the authors used a multi-start global optimization algorithm in order to design optimum laminated composite plates with maximum stiffness. Many researchers have been studying these problems for many years [38, 39, 47] and they have focused their efforts on minimizing structure weight objectives. The need of using global minimization techniques is enormous, as the increased number of design variables, the nonlinear way in which surface deformations are expressed and other similar elements make conventional minimization techniques insufficient. As a result, optimal design was limited and advanced optimization methods were required. Practical application to the above-mentioned problem can be found (perhaps the most known) multi-start global optimization algorithm of Snyman-Fatti [71]. For this purpose, this algorithm has been expanded in order to provide a solution to the optimal design problem. In the paper [40], the authors aim to the optimal shape selection the laminated surface in order to achieve the maximum flexural stiffness of the plates. In addition, the plate layer and the density of each layer are taken into account. Thus, the problem is expressed in a mathematical way as follows:

Problem 1 For the strain energy of a plate F, find $\min F(x)$ where $x = (x_1, x_2, \ldots, x_n)^\top$ is the vector of ply orientations such that $0^\circ \leqslant x_i \leqslant 180^\circ$ and $\sum_{i=1}^{n} t_i = t$, where t_i denotes the thickness of i-th ply of the plate, while t represents the total thickness and n represents the total number of the layers.

To solve the above problem, the authors applied the Snyman-Fatti method and their results were very encouraging. Specifically, they applied the method to the following problems, creating plates of 4 to 16 layers: (a) *Optimum ply orientations of centrally loaded plates*, (b) *Optimum ply orientations of uniformly loaded plates*, (c) *Optimum ply orientations of plates subjected to combines loads*, (d) *Reduction of displacements by using optimum design* and (e) *Effect of Young's ratio upon optimum ply orientation*.

An interesting case where differential equations are encountered is the min-max nonlinear problems. In the paper [83], the authors presented three algorithms, which by solving initial value problems for ordinary differential equation systems, they tackled four well-known numerical min-max problems. In addition, for two of these algorithms, they have shown global convergence criteria. Specifically, the authors studied the following problem:

Problem 2 For a continuously differentiable function F and parameters x, y, find:

$$\min_x \max_y F(x, y),$$

where $x \in X \subseteq \mathbb{R}^s$ and $y \in Y \subseteq \mathbb{R}^r$ are s and r dimensional decision vectors and:

$$X = \{x \in \mathbb{R}^n : x^l \leqslant x \leqslant x^h\},$$
$$Y = \{y \in \mathbb{R}^m : y^l \leqslant y \leqslant y^h\},$$

where $x^l, x^h \in \mathbb{R}^n$, $y^l, y^h \in \mathbb{R}^m$ and $x^l < x^h$, $y^l < y^h$.

According to the authors, the x, y parameters are renewed depending on the solutions of the differential equations. Additionally, the following assumption has been made: a min-max solution of F exists and is unique, since appropriate constraints regarding the decision variables have been carried out [83]. It is noteworthy that the information of the gradient, where it was necessary to be calculated, was done in a numerical way by the provided algorithms. In the first problem, all three algorithms easily found the desired solution. The differences observed on the trajectories, however, suggest the need for further study of the correlations in the obtained results. In the second problem, the authors note that the convergence of the algorithm may not lead to a good solution. In the third problem, they showed that the right determination of the conditions plays an important role. Bad determination leads to non-convergence of the algorithm. Finally, through the solution of the last problem, they showed that the trajectories can move within or over the boundary of the set of constraints.

A new algorithm, called *DYNAMIC-Q* algorithm, is presented in [75]. This method was an easy-to-use and reliable algorithm of tackling constrained optimization problems [60]. It is essentially a variation of an existing algorithm [69, 70] based on dynamic search trajectories which addressed unconstrained optimization problems. Thus, the aforementioned algorithm has been modified in such a way in order to handle constraints. In particular, a dynamic penalty parameter is introduced in the algorithm and as a result it can solve a constrained optimization problem [60]. This new method has been applied to structural optimization problems, such as minimum weight structures of trusses and frames. Optimization problems of this kind are very difficult to be solved, since a large number of parameters have to be taken into account, such as trend, displacement, weight etc. For this reason the direct application of trajectory methods would not be efficient. Thus, the proposed method was applied in order to solve subproblems and therefore the solution of the original problem becomes considerably less expensive. In detail, the penalty function which occurs in the *DYNAMIC-Q* is formulated as follows:

$$P(x) = F(x) + \sum_{i=1}^{m} a_i \, G_i^2(x) + \sum_{j=1}^{s} b_j \, H_j^2(x), \tag{48}$$

where $a_i = 0$ if $G_i(x) \leqslant 0$, otherwise $a_i = \rho_i$ if $G_i(x) > 0$. In many cases, for convenience $\rho_i = b_j = \mu > 0$ is taken, where μ denotes a large number. Under certain conditions, the minimum of $P(x)$ agrees with the minimum of $F(x)$ concerning the following minimization problem with constraints:

Problem 3 For a function $F(x)$, where $x \in \mathbb{R}^n$ find the min $F(x)$ such that:

$$\begin{aligned} G_i(x) &\leqslant 0, \quad i = 1, 2, \ldots, m, \\ H_j(x) &= 0, \quad j = 1, 2, \ldots, s, \end{aligned}$$

where F, G and H represent scalar functions of x.

Since large values of the parameter μ can create a difficult to handle problem, the following strategy was proposed. The penalty parameter μ is more advantageous to increment gradually, as the trajectory evolves, until a predetermined value is reached. Therefore, initially set $\mu = \mu_0$, define a factor c (which is usually a small number close to 1) and a maximum value $\mu_{\max}$ too. Thus, at each step, we set $\mu_k = c\mu$ until the inequality $\mu_k > \mu_{\max}$ is satisfied. Then we set $\mu_k = \mu_{\max}$ and continue the process until convergence is achieved.

The authors of the paper [75] observed that adopting this issue a better trajectory is produced. In addition, a faster convergence towards the $x^\star$ region is achieved, compared to the non-stepwise increase of the μ parameter. The difficult issue that is created and makes the problem computationally expensive is the following. The authors face a structural optimization problem that requires the evaluation of gradient for both the objective and constrain functions. This raises the computational cost of the problem. A solution to this problem is given by the succession of quadratic subproblems. In summary, the following constraint formula is used:

$$G_j(x) = G_j(x_k) + \nabla G_j(x_k)^\top (x - x_k) + m_j^k \|x - x_k\|^2, \qquad j = 1, 2, \ldots, m, \tag{49}$$

where m determines the number of constraints and the coefficient m_j^k has to be appropriately calculated for each constraint. The last formula with the objective function $F(x)$ constitutes the subproblem. The stopping criterion which is adopted in this process was the following:

$$\frac{\left|F(x_{k+1}^\star) - F(x_k^\star)\right|}{F(x_{k+1}^\star)} \leqslant e_r, \tag{50}$$

where $x_{k+1}^\star$, $x_k^\star$ denote the last two subproblems solution and e_r represents a predefined small, positive parameter. It is worth noting that if we put $m_j^0 = 0$ then the problem turns into a linear programming problem. The authors, in all the experimental tests they conducted, suggested $m_j^0 = 0.01$. The proposed method was tested in 10 problems: 5 truss problems and 5 frame problems. Moreover, the problems were different in terms of dimension and number of variables. The *DYNAMIC-Q* algorithm compared with the *sequential quadratic programming (SQP)* method. The results showed that the provided scheme was competitive, reliable (mainly for computationally expensive problems) and solved the problems quickly and efficiently.

In the paper [66] the authors presented a trajectory method for solving unconstrained optimization problems and specific unconstrained nonlinear programming problems. Utilizing some properties of the objective functions and a special system of ordinary differential equations, they produced trajectories that gave very good results. In addition, an important advantage of their approach is the usage of gradient information and not the value of the objective function itself. Moreover, it is worth pointing out that the computational cost was not very high, as the proposed method did not evaluate high derivatives. Next, we present the initial value problem [12, 13] that they used in order to produce the trajectories, as well as the problem that they approached. The unconstrained nonlinear problem was of the following form:

Problem 4 Assume the function $F : \mathbb{R}^n \to \mathbb{R}$, $F \in \mathbf{C}^2(\mathbb{R}^n)$ and let $S_a = \{x : F(x) \leqslant F(a)\}$ be a compact set $\forall\, a \in \mathbb{R}^n$. Find $\min\limits_x F(x)$ by solving the following initial value problem:

$$\begin{cases} \dot{x}(t) = \dfrac{-\nabla F\big(x(t)\big)}{\big\|\nabla F(x(t))\big\|_2^2\, s(t)}, \\ x(0) = x_0, \end{cases} \tag{51}$$

where $t \geqslant 0$, $s \in \mathbf{C}^1$ and $\nabla F\big(x(0)\big) \neq 0$.

Next, we present some very interesting theorems, which demonstrate important properties of the method proposed by the authors [66].

Theorem 4 *Assume that $I = [0, t)$ is a maximal interval such that the above initial value problem (51) has a unique solution within this interval. This solution can be extended continuously to the interval $\tilde{I} = [0, t]$ and $\nabla F(x(t)) = 0$.*

Theorem 5 $F(x(t)) = F(x_0) - \int_0^t s(\tau)d\tau = G(t)$.

Theorem 4 shows the existence and uniqueness of the solution of the initial value problem (51), while Theorem 5 suggests that the function G can be found for different choices of s. Moreover, although the solutions of (51) are unknown, the functional values of F can be given using the function G. This element is used by the authors in order to indicate that the step size control can be performed. In addition, the reader is referred to [66] in order to study alternative differential equations with equivalent results except of the advantage of the function G. The authors through the next theorem showed that the well-known steepest-descent method with Armijo-Goldstein step size control is equivalent to the Euler method for $s \equiv 1$ using the G function for the step size control. Specifically, if h denotes the step size, then it can be selected using the information of G with the following way: For a parameter δ, where $0 < \delta < 1$ there can be a step size $h > 0$ such that:

$$F\big(x_{\text{new}}(h)\big) - F\big(x(0)\big) \leqslant \delta\big[G(h) - F\big(x(0)\big)\big], \tag{52}$$

where the $x_{\text{new}}(h)$ is calculated using the Euler method applied on the initial value problem (51). Furthermore, they gave the following theorem:

Theorem 6 *Suppose that the Euler's method is applied to the initial value problem (51). Then, for functions similar to those of Probllem (4) and for $s \in \mathbf{C}^1$, $s(t) > 0$, $\forall t \in \tilde{I}$ the step size control (52) is possible.*

The interested reader is referred to [66, 88] for details of the proofs of the aforementioned theorems. Furthermore, the authors tested the proposed method with other known trajectory-following methods. They presented the total number of iterations that are required for each problem. In particular, they used a modified implicit *Runge-Kutta* method of 4/5 order. The experiments they conducted include well-known benchmark functions such as: (a) *Rosenbrock's function*, (b) *Wood's function*, (c) *Powell's function* and (d) *exponential form functions*. The experimental results showed that the provided method was competitive.

In the paper [72] a modification of the Snyman-Fatti algorithm [71] is presented in order to tackle non-convex constrained global optimization problems. The unconstrained version of the algorithm was successfully applied [40] in order to solve optimization problems encountered in structures. The basic idea of the proposed method was: (a) initially, the authors applied the traditional *Snyman-Fatti* method (the unconstrained algorithm) to calculate the global minimum of a penalty function, similar to (48), (b) in the next step, the recognition of all the limitations concerning the solution of the penalty function took place and (c) they applied the Snyman-Fatti algorithm in order to minimize a least squares function. The authors, in order to check the performance of the proposed method, which called *Snyman Fatti CONstrained (SFCON)* algorithm, they used a set of known constrained problems [24]. The obtained results showed that the SFCON method is highly effective, since it founded the solutions of all the tested problems (the method tested in 10 problems, one of

which has 5 special cases). The accuracy of the method was satisfactory and despite the fact that the number of function evaluations was high, the CPU time was not so large.

Another variation of the *Snyman-Fatti* algorithm was presented in the paper [30]. The authors pointed out that in the modified version, significant improvements in accuracy and effectiveness are obtained. In particular, they distinguish some basic stages: (a) the *global phase* of the algorithm where the algorithm tries to approach a neighborhood of minima. Thus, one can claim that the algorithm converges to the minimum although, initially, the global minimum of the objective function is not reached, (b) then, the *local phase* of the algorithm emphasizes to a higher accuracy. Thus, since the trajectory, firstly, is moved to a relative low minimum is possible to locate the smallest of the minimum points. Additionally, parameter selection plays a key role and the test of bound violations is required. Furthermore, except of the above phases, it is necessary to check and change some specific parameters such as the time step Δt and the parameter a which is appeared in the original algorithm. An important detail is also the fact of breaking a boundary by a trajectory. In this case the component of $x(t)$ that violates the bound is regenerated within the region of interest in a random way. The authors tested the proposed method, called the *SF-M* method, with other known global optimization methods [30]. In addition, the proposed method was tested on optimization problems of nontrivial orthotropic membrane, plates and shell structures. The experimental results showed that the method is robust and accurate, while its performance is better than the compared methods, especially when functions with a large number of variables or a large number of local minima are considered.

In [68] the authors presented an algorithm, called *nonlinear optimization via external lead (NOVEL)* algorithm. This algorithm achieves global optimization and is applied in learning process of an *artificial neural network*. Through their study they concluded that in order to build a good search process, it must meet two basic characteristics: (a) to be able to use the gradient information in order to perform a good local search in the search-area and (b) it can be unleashed by local minima domains in order to search for the global minimum of the objective function. Local minimization methods have been tested effectively in some cases [6, 16]. Their drawbacks, however, such as the difficulty of handling flat surfaces, have led to the need of global minimization methods for the learning process. This, of course, does not mean that one can blindly "trust" a global minimization method, whether it is a deterministic method or a probabilistic one. The reader can easily understand that methods belonging to this class have their own disadvantages too. For example, when the search space is too large, the convergence of the method may be terribly slow or, more specifically, the gradient information is very useful but it cannot be used in simulated annealing. For this reason, *NOVEL* algorithm is a hybrid scheme that combines global and local features. In more detail, it uses a trajectory method in order to escape from a local minima area and local descents in order to locate these local minimum regions. In summary, the algorithm is distinguished by the following features: (a) it searches the solution space, (b) it locates regions where is possible to exist minima points and (c) it identifies local minima. In particular, regarding the first feature of the algorithm, the exploration of the solution space is carried out by a continuous terrain independent trace which is not trapped in local minima. Then local gradient is used and relied on the trace in order to be unleashed by a local minimum. Finally, an initial point from each promising local region is selected. Then a descent algorithm takes these points as initial values and local minimum are located. In conclusion, *NOVEL* algorithm utilizes a differential equation to conduct global search and locate regions that contain local minima. Then, a local search method is applied in order to calculate these points. The inputs to this algorithm, for example a conjugate gradient method,

are given by the trajectory method. At this point, it is worth mentioning some issues of the global search phase more analytically. During the global search phase of the algorithm an ordinary differential equation of the following form is used:

$$\dot{X}(t) = P\left(\nabla_X f\left(X(t)\right)\right) + Q\big(T(t), X(t)\big), \tag{53}$$

where t is an autonomous variable, T denotes a trace function and P, Q are general nonlinear functions. The last equation consists of two terms:

(a) $P\left(\nabla_X f(X)\right)$ represents the trajectory attraction made by the gradient in order to locate a local minimum, while
(b) $Q(T, X)$ denotes the trace function which guides the trajectory out of the local minimum region.

It is worth paying attention to the following issue. In order to explore the search space, the authors adopted the search from coarse to fine. This was done in order to avoid the space fragmentation in too many sub-spaces. Especially for many dimensions this would be impractical. Thus, the authors created a non-periodic analytical trace function [68]. In addition, two ways were used in order to solve the differential equation: (a) a solver called *Livermore solver for ordinary differential equations (LSODE)* and (b) a finite-difference equation solver. The authors conducted extensive experiments to test the performance of their algorithm. They test the *NOVEL* algorithm on the well-known problems: (a) the *two-spiral* problem, (b) the *sonar* problem, (c) the *vowel recognition* problem, (d) the *10-parity* problem and (e) the *NetTalk* problem. They compared their method with the following well-known global optimization algorithms: (a) *simulated annealing*, (b) *evolutionary algorithms*, (c) *sascade correlation* with multistarts, (d) *gradient descent* with multistarts and (e) *truncated Newton's* method with multistarts. The results have shown that the usage of difference-equation solver is better in terms of speed than the *LSODE* solver. However, the *LSODE* solver provided more quality solutions. In summary, this new method improved the learning of feed forward neural networks, while it worked very well in optimization of general high-dimensional nonlinear functions. For more details concerning the performance of the *NOVEL* algorithm for each problem, the reader is referred to [68].

In [73] an algorithm with low storage requirements for handling constrained optimization problems has been proposed. This algorithm is based on an existing dynamic trajectory method and is appropriate for functions with many variables. In particular, the proposed algorithm named *dynamic-Q* has the advantage that no Hessian information is required. Moreover, successive subproblems are produced with the construction of spherically quadratic approximations. The authors used the well-known *leap-frog algorithm* in order to solve these subproblems. The provided method compared to the Sequential Quadratic Programming method presented competitive results.

In the paper [44] a hybrid global optimization method based on the well-known multistart algorithm [71] has been proposed. The authors used an algorithm from the field of evolutionary computation, in particular a *differential evolution (DE)* algorithm [80]. Thus, by using a metaheuristic technique generate a population of points in order to search the space. During the DE phase take place two widely used operators namely *mutation* and *recombination*. The basic idea was to identify the auxiliary trajectories in a more advanced way than the defined way in the original work of Snyman and Fatti. Specifically, the procedure starts with a randomly selected point x_0^1, and by using the method proposed in [71] generated the first trajectory T_1. The initial point is introduced in a population P which is empty at the beginning of the process. The trajectory T_1 is terminated at the point x_n^1. This point in turn, is merged in population P. At this point of the algorithm, the auxiliary

trajectory x_0^2 starts, as above. As it is well-known, the DE algorithm cannot be started yet, since the population P must consist of at least four individuals. For this reason, the original design of the auxiliary trajectories is followed, until the population P obtains the necessary number of individuals. Therefore, the DE algorithm starts when the following individuals are merged into the population:

$$P = \left\{x_0^1,\ x_n^1,\ x_0^2,\ x_n^2\right\}. \tag{54}$$

At this stage, the operators of DE algorithm, *mutation* and *recombination* are applied. As a result, four new individuals are born. Thus, the *selection* task is activated and a whole new population is generated. The best individual of the new population, let P_2, is chosen in order to create the trajectory T_3 (T_1 and T_2 have already appeared). The x_0^3 is entered into the population P and the remaining points of P_2 are abandoned. The process continues in the same way and new trajectories are constantly produced and similarly are included in the population P. An issue arises when the population reaches the maximum number of individuals that it can include. This constitutes a problem because it is possible an individual to be born which has a lower function value than an already included point. If this happens, (if the population reached its maximum size), then existing individuals are classified according to their function values. If there is a new point with better features, it replaces the worst individual. The process we just described was the main contribution of the authors, in order to improve the performance of standard Snyman-Fatti method. The authors conducted experimental tests using ten well-known benchmark functions of different dimensions. Particularly, they used in their test the following functions: (a) the *Rosenbrock function* ($D = 2$), (b) the *Freudenstein-Roth function* ($D = 2$), (c) the *Hellical valley function* ($D = 3$), (d) the *Levy No. 8 function* ($D = 3$), (e) the *Wood function* ($D = 4$), (f) the *Watson function* ($D = 6$), (g) the *Hyper-Ellipsoid function* ($D = 6$) and (h) the *Rastrigin* function ($D = 6, 10, 15$). It is noteworthy to mention that the proposed method outperforms the standard Snyman-Fatti method and in some cases, the improvement rate reached the 90% or more. Moreover, there were cases where the hybrid method achieves convergence for which the plain Snyman and Fatti method does not converge.

Optimization methods using dynamic search trajectories become more interesting due to their large number of applications. Trajectory methods have been applied successfully on many applications related to *computational intelligence*. Next, we briefly describe these methods for the *artificial neural network (ANN)* training task. As it is reported in [53] during the training process of an ANN, it is required the minimization of a function, known as *error function*. This makes the training process equivalent to an optimization problem. The authors in their paper [54] proposed a new way of minimizing the error function, based on the well-known method of Snyman-Fatti [71]. Assuming that the error function of the network is denoted by E, they consider the differential equation:

$$\ddot{x} = -\nabla E\big((x)\big), \tag{55}$$

where x represents the vector of the synaptic weights of the ANN. The authors studied a particular class of neural networks, the *multilayer feedforward neural networks (MFNNs)*. They made a variation on the original dynamic trajectory search method of Snyman-Fatti and by applying it to the training of an *MFNN* task they obtained very good results. It is worth mentioning that through the conducted study the authors concluded that any numerical method for solving ordinary differential equation can be applied to a neural network

training algorithm. The proposed method differs from the original Snyman-Fatti method and applies four additional steps. Below we briefly present how this new method operates:

A trajectory starts its path and the values of the error function are stored. If the value of error function E decreases as the trajectory moves, then the integration step increases according to a parameter ζ. This takes place in order to accelerate the convergence process to the minimum of the function. On the other hand, if the ratio of the current functional value with respect to the previous one overcomes a parameter β, then the last visited point by the trajectory is maintained. Next, a new trajectory starts from this point. If the step is not efficient enough, the proposed method applies the well-known "Armijo rule" [2] and the step is halved. The authors have pointed out that the following relation should be taken into consideration:

$$\frac{E(x_{t+1})}{E(x_t)} > \beta. \tag{56}$$

Remark 2 There are important issues that have to be faced during the neural network training, namely, the selection of (a) the *topology (architecture)* and (b) the learning algorithm. It is well-known that if this selection is inappropriate, then the applicability of the neural network may not be effective. In general, this is not a trivial problem, and its solution has been concerned the scientific community for many years. Many methods have been proposed in the literature in order to achieve local minimization. However, it is up to date and very useful to modify existing minimization methods in order to tackle the global optimization problem.

To test the above mentioned *modified Snyman-Fatti* method [54], the authors To test the above mentioned conducted experiments using three well-known problems from the "proben1 dataset", namely the problems: (a) *Cancer1*, (b) *Diabetes1* and (c) *Heart1*. The network topologies (architectures) used were: (a) the $9-4-22$, (b) the $8-2-2-2$ and (c) the $35-8-24$. The proposed method compared with well-known variations of *back propagation (BP)* training algorithm namely (a) the *back propagation with momentum (MBP)*, (b) the *second order momentum (SMBP)*, (c) the *adaptive back propagation (ABP)* (using the adaptive scheme proposed by Vogl [84]), (d) the *parallel tangents method (PARTAN)*, (e) the *scaled conjugated gradient (SCG)*, (f) the *resilient back propagation (RPROP)* and (g) the *improved resilient back propagation (iRPROP)* algorithm. The experimental results have shown that, regarding the Cancer1 problem the proposed method was ranked third in terms of the number of function evaluations that are required. In addition, it has achieved the second best performance for the classification error. Concerning the Diabetes1 problem, it was ranked third regarding the number of function evaluations and the classification error as well. Finally, for the Heart1 problem, it has achieved the best second and third performance respectively. In conclusion, although the proposed modified Snyman-Fatti method was not the best one for all the tested problem, it was competitive and it was on the top three methods with the best performance in all the tested problems.

In [53] the authors used numerical methods for solving initial value problems [12, 13] in order to solve the inherent optimization problem that exists in neural network training. The class of the methods that they proposed is a subclass of dynamic search trajectory methods. Specifically, in order to tackle this problem they solved the following ordinary differential equation:

$$\dot{x} = -\nabla E\big((x)\big), \tag{57}$$

where x represents the weight of the network and E determines the *error function* of a *multilayer feedforward neural network (MFNN)*. The authors for solving the corresponding initial value problem:

$$\dot{x} = -\nabla E\big((x)\big), \quad x(0) = x_0, \tag{58}$$

applied numerical Runge-Kutta methods. Specifically, they applied the second order and two stage Runge-Kutta methods:

$$x_{n+1} = x_n + a_1 k_1 + a_2 k_2, \quad n = 0, 1, \ldots \tag{59}$$

where:

$$\begin{aligned} k_1 &= h \nabla E(x_n), \\ k_2 &= h \nabla E(x_n + b_2 k_1) = h \nabla E(x_n + h b_2 \nabla E(x_n)), \end{aligned} \tag{60}$$

and $h > 0$ is a given step size. If the values of the parameters a_1, a_2 and b_2, fulfill a system of algebraic equations then the corresponding obtained methods are considered as second order Runge-Kutta methods. Also, an infinite number of second order Runge-Kutta methods can be constructed [12, 13]. The authors set the following values for the parameters a_1, a_2 and b_2, in order to obtain the corresponding RK1, RK2 and RK3 solvers:

(a) *RK1 method*: $a_1 = 0$, $a_2 = 1$ and $b_2 = 1/2$,

(b) *RK2 method*: $a_1 = a_2 = 1/2$ and $b_2 = 1$,

(c) *RK3 method*: $a_1 = 1/4$ and $a_2 = b_2 = 2/3$.

The authors tested the proposed methods on three well-known problems from *proben1 database*. Namely the problems: (a) the *XOR*, (b) the *Coder-Decoder* and (c) the *Cancer1* classification problem. The corresponding neural network architectures were: (a) $2-2-1$, (b) $4-2-4$ and (c) $9-4-4-2$. The proposed schemes have been compared with the well-known Backpropagation (BP) family of methods. The experimental results have shown that the proposed Runge-Kutta methods outperformed the BP family. Specifically, regarding the *XOR* problem, the *RK1 method* required less number of function evaluations from all the others, while the lower standard deviation was achieved by the *RK3 method*. Concerning the second problem, all the proposed Runge-Kutta methods required less function evaluations than all the other methods and the only competitive method was the *adaptive back propagation* method. Finally, regarding the *Cancer1* problem, the least computational demanding method was the *RK2 method*, while the least classification error achieved by the *RK3 method*.

The authors continued their effort of addressing the minimization problem that occurs for the training of a neural network with their work in [55]. In this paper they proposed a new method for dynamic search trajectories using the *Stoermer rule* [61], with respect to the initial value problem:

$$\ddot{y} = f\big(x, y(x)\big), \quad y(x_0) = y_0, \quad \dot{y}(x_0) = z_0. \tag{61}$$

In particular, they solved the following initial value problem:

$$\ddot{x} = -\nabla f(x), \quad x(0) = x_0, \quad \dot{x}(0) = 0, \tag{62}$$

and compared their scheme with the *BP* family of methods, as they did in their previous works. Next, we briefly review the functionality of the above-mentioned rule and how it was applied by the authors.

The *Stoermer's rule* is described by the following formulas:

$$\begin{aligned} y_1 &= y_0 + h\left[z_0 + \tfrac{1}{2}hf(x_0, y_0)\right], \\ y_{n+1} &= 2y_n - y_{n-1} + h^2 f(x_0 + nh, y_n), \quad n = 1, 2, \ldots, m-1, \\ z_m &= y_m - \tfrac{y_{m-1}}{h} + \tfrac{1}{2}h\, f(x_0 + H, y_m), \end{aligned} \tag{63}$$

where H represents the total step that has to be used in m sub-steps. Thus, each substep is of length $h = H/m$. The final value, z_m is $\dot{y}(x_0 + H)$. In [32] it has been shown how to rewrite the above (63) in order to reduce the roundoff error using the quantities:

$$\Delta_n = y_{n+1} - y_n. \tag{64}$$

Thus, using this approach the method obtains the following form:

$$\begin{aligned} \Delta_0 &= y_1 - y_0 = h\left[z_0 + \frac{1}{2}h\, f(x_0, y_0)\right], \\ y_1 &= y_0 + \Delta_0, \\ \Delta_n &= \Delta_{n-1} + h^2 f\big(x_0 + nh, y_n\big), \quad n = 1, 2, \ldots, m-1, \\ y_{n+1} &= y_n + \Delta_n, \quad n = 1, 2, \ldots, m-1, \\ z_m &= \frac{\Delta_{m-1}}{h} + \frac{1}{2}h\, f\big(x_0 + H, y_m\big). \end{aligned} \tag{65}$$

As in their previous works that are analyzed above, the proposed method was encountered with the *BP* family of methods. The problems that used were: (a) *Cancer1*, (b) *Diabetes1* and (c) *Heart1* with fixed training and test sets. The results showed that the provided method achieved less function evaluations in comparison with all the others, regarding the Cancer1 and Diabetes1 problems. Regarding the classification error the Stoermer method ranked second behind the *adaptive back propagation* method. Finally, regarding the Heart1 problem, the proposed method achieved the first and second performance respectively.

3 Recent dynamic search trajectory methods and their applications

In this section, we present the most recent optimization methods based on the dynamic search trajectories. Specifically, we provide the methods which have been developed over the last ten years.

Significant efforts using trajectory methods have been made in order to tackle difficult problems such as multi-objective optimization, as well as large scale global optimization problems [81, 82]. The authors in these two papers, proposed a new algorithm, named *multiple trajectory search (MTS)*. The basic concept of this algorithm was that many trajectories search the solution space and tree local search methods take place. As a result, depending on the form of the solution space, the local methods could be activated, whenever it was necessary to locate either a local or global minimum. The proposed method was compared mainly with evolutionary methods [42, 45] and the experiments they conducted have shown profitable results.

At this point, we would like to underline the contribution and the importance of the past methods, which greatly determined the current approaches. An element that reveals their contribution is both the improvements that have been made regarding these past algorithms and the reassessments made concerning the original works in order to test the efficiency of the methods in additional problems. In the paper [74], Snyman and Kok reassessment the

well-known *multi-start global minimization algorithm with dynamic search trajectories* [71] and compare it with Storn and Price *differential evolution algorithm* [80] on an extensive set of problems. Since these two algorithms are well-known and the first one has been extensively presented in Section 2, we will focus only on some very interesting outcomes derived from the experiments conducted by the authors. Initially, their comparison included 48 known problems with functions of different dimensions and of wide range. Specifically, 2-dimensional functions up to 20-dimensional functions with continuous variables were included in the experiments. The experimental results have shown that *Snyman-Fatti method* reached very good results, since it achieved in 43 out of 48 cases (failed to solve 5 problems – four 9-dimensional functions and one 10-dimensional function). In addition, for more than half of the tested problems the method achieved 100% success rate. In the cases where the success rate was not particularly high, the authors propose as a correction to increase the minimum number of iterations. Considering the performance of the methods regarding the computational burden (minimum number of function evaluations), it is readily understood that the two methods are not comparable. This is so, because the *Snyman-Fatti method* requires the gradient function evaluation, while *DE algorithms* by their nature do not require gradient function evaluation. The authors, nonetheless, repeated some experiments, using the analytical expression of the gradient. As it was expected, the CPU time reduction rate was remarkable. Also, it is worth noting that there were 8 problems in which Snyman-Fatti method required fewer function evaluations. The authors also tested this method in the case where the global minimum is known. The experimental results showed that the Snyman-Fatti method succeeds in finding the global minimum in 44 problems. Finally, the nature of the original Snyman-Fatti method suggests that the problem of minimizing potential energy will be successfully approached by this method. In fact, the authors tested the *Snyman-Fatti method* in the known problem [87] and the results again have shown that the Snyman-Fatti method is highly competitive compared to DE algorithms. A further advantage is that Snyman-Fatti method can be easily modified in order to solve constrained optimization problems.

An interesting application of the *Snyman-Fatti algorithm* (in a further improved version) was presented in the paper [20]. This paper examines a well-known problem in the *steel structures* concerning the *stiffened plates*. Let us briefly describe this problem. In many structure problems [19], such as in building floors, platforms or in building a bridge, stiffened plates are used. An important issue that is inherent in these problems is the optimal design of welded square stiffened plates and of course, the minimization of a series of parameters. In particular, in this problem we have the following constraints:

(a) The stress in the base plate and, also, in stiffeners.
(b) The stress on the deflection of edge stiffeners and, also, of internal stiffeners.

As a result, we have to minimize the cost function that includes the following parameters: (a) the material, (b) the welding and (c) the painting cost. The cost function is derived according to the fabrication sequence. Moreover, we have to take into account the design parameters, such as: (a) the base plate thickness, (b) the dimensions of edge and internal stiffeners and (c) the number of internal stiffeners. These constraints lead to a function where the appropriate optimization is required. For more details we refer the interested reader to [20]. Obviously, this is an optimization problem with several constraints. Thus, by taking a suitable transformation, the initial problem is properly modified to an unconstrained minimization problem, in order the Snyman-Fatti method can be applied to provide the solutions. The authors compared the *Snyman-Fatti method* to the well-known *particle*

swarm optimization (PSO) algorithm. The obtained results have shown that *Snyman-Fatti method* is competitive, robust and effectively accurate for the optimal design task.

Next, we present another application which is related to dynamic search trajectories and neural networks. Specifically, in the paper [9], the authors studied a specific kind of functions named *non-convex scalar functions*. Particularly, their approach is based on the ordinary differential equation:

$$\ddot{x} + \gamma\, \dot{x} + \nabla f(x) = 0, \tag{66}$$

where $\gamma > 0$ is a positive scalar number, $f \in \mathbf{C}^1$ is an objective function with a unique global minimum related to a neural network algorithm known as *continuous-time algorithm* [7]. The main concept was the construction of a new continuous-time method based on the well-known *conjugate gradient* method. Using this new scheme, the trajectories that are produced by the solution of the ordinary differential equation, may be released from local minimum. The proposed algorithm, called *controlled conjugate gradient (CCG)* network is described by the following system of ordinary differential equations:

$$\begin{aligned} \dot{x} &= a(x,u)\, u, \\ \dot{u} &= -\nabla f(x) - b(x,u)\, u, \end{aligned} \tag{67}$$

where a, b represent functions. This method is based on the *control Liapunov function (CLF)* which is described in [8]. The provided method has been compared with the *heavy ball with friction (HBF)* network [59]. The experiments conducted by the authors include the following test functions: (a) *MATLAB's peaks* function, (b) *Rastrigin's* function, (c) *Ackley's* function and (d) *Griewank's* function. The majority of *CCG* algorithms converged faster to the minimum than the *HBF* algorithm. Furthermore, the authors have pointed out that if the adjustment of the parameters become very careful, then the CCG family of methods escape from a local minimum. Despite the fact that the results were promising, the authors pointed out that more experimental test are needed. For this reason they proposed to test the *CCG* networks using other known trajectory methods, such as [44] and [71].

The applications of methods using dynamic trajectories are not limited to the field of Physics, Mathematics or Mechanics, as we have already discussed in the introduction of this paper. In [5], the authors provided an open code algorithm for greedy search in order to tackle the maximization of the gain in information about the desired goal. Firstly, they received the appropriate equation of motion such that the trajectories move forward to the solution. The authors have focused on $2-$dimensional and $3-$dimensional cases.

In [76] several reasons where the gradient-based methods are inappropriate for handling real-world problems are presented by the authors. Some of these reasons are the following:

(a) Often the problems under consideration are very expensive to evaluate, requiring, for example, the time-consuming finite element analysis of a structure.
(b) Sometimes discontinuities or/and noise are presented.
(c) In many cases multiple local minima are observed.
(d) There can be regions in the design space where the function is not defined and
(e) The number of design variables may be huge.

The aforementioned difficulties lead researchers to the development of new optimization algorithms. The paper [76] gathers the results that have been achieved mainly by Snyman and his colleagues for handling the above difficulties in unconstrained optimization as well as in constrained optimization problems.

In Table 1 a citation analysis using the most well-known and most cited methods[1] is provided. This presentation briefly exhibits information about the methods that use dynamic trajectories for solving an optimization problem. In addition, it offers to the reader a direction on which methods have gathered the interest of the researchers the most.

4 Families of dynamic trajectory methods derived from the Runge-Kutta methods

In this section we present dynamic trajectory methods for global optimization. These methods are derived from the Runge-Kutta methods for solving ordinary differential equations. We consider the autonomous initial value problem:

$$\dot{y}(x) = f\big(y(x)\big), \quad y(x_0) = y_0, \quad f : \mathbb{R}^N \to \mathbb{R}^N. \tag{68}$$

Remark 3 In general, an autonomous differential equation is a differential equation which does not explicitly depend on the independent variable. In the case where the independent variable is the time is called time-invariant. Various laws of nature are expressed as autonomous systems where the independent variable is usually assumed to be the time. This is so because the laws of nature that hold at the present time are considered to be identical to those for any point in the past or in the future. Furthermore we point out that it is always possible to write a non-autonomous equation in an equivalent autonomous form [13].

A well known method for solving the autonomous initial value problem (68) is the following Euler's method which is the simplest RungeKutta method [12, 13, 31, 37, 43]:

$$y_{n+1} = y_n + hf(y_n), \quad h = x_{n+1} - x_n. \tag{69}$$

The above method is considered as first-order and one-stage Runge-Kutta method. The error in a single step is proportional to the square of the step size $h > 0$ (the error in a single step behaves like $O(h^2)$). In this case, the corresponding Butcher's tableau is the following [13]:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

It is obvious that, if we apply the Euler method (69) to the following autonomous initial value problem:

$$\dot{x} = -\nabla F(x), \quad x(0) = x_0, \tag{70}$$

we obtain the traditional Cauchy's method for the optimization of the objective function $F : \mathbb{R}^N \to \mathbb{R}$:

$$x^{k+1} = x^k - h\,\nabla F\big(x^k\big), \quad k = 0, 1, \ldots, \tag{71}$$

We can obtain more accurate results for solving (68) by using second-order two-stage Runge-Kutta methods for which the error in a single step behaves like $O(h^3)$. For example we can use the following well-known and widely used Runge-Kutta methods:

(a) the *Euler-Cauchy method* which is related to the midpoint quadrature method,
(b) the *Heun method* which is related to the trapezoidal rule quadrature formula and
(c) the *Ralston method* which exhibits a minimum local error bound.

[1] Source: Google Scholar

Table 1 Brief description, applicability and number of citations of the references related to the trajectory methods presented in Sections 2 and 3

Title	First-order ODEs	Second-order ODEs	TC	CpY	Brief description
"Widely convergent methods for finding multiple solutions of simultaneous nonlinear equations"	(yes)	(no)		9.00	In this paper, the author tackles the problem of finding the solution of a system of non-linear equations. In particular, the proposed method can identify multiple roots and, in some cases, achieves global convergence [11]
"Generalized descent for global optimization"	(no)	(yes)	400	10.53	Griewank has presented some basic properties of the trajectory that are required to be fulfilled. In addition, the authors formulates and proves a theorem, which reinforces the opinion that the solution of the differential equation under consideration is transformed into the trajectory of the steepest descent [27]
"A new and dynamic method for unconstrained minimization"	(no)	(yes)	181	4.89	A practical localization method of local minima was proposed, mainly for function that the first derivative can be easily estimated. This method was aimed at locating the minima of multivariable functions. Snyman's method responds very well in cases where the starting point is not close to the local minima [69]
"A multi-start global minimization algorithm with dynamic search trajectories"	(no)	(yes)	150	4.69	The authors produced a global minimization method using a multi-start algorithm. Under appropriate modifications to the searching trajectories, the authors managed to broaden the convergence regions regarding the global minimum of the objective function [71]

Table 1 (continued)

Title	First-order ODEs	Second-order ODEs	TC	CpY	Brief description
"Evolutionary operators in global optimization with dynamic search trajectories"	(no)	(yes)	14	0.88	A hybrid global optimization method based on the well-known multi-start algorithm [71] has been proposed. The authors used an algorithm from the field of evolutionary computation, in particular the *differential evolution (DE)* algorithm [44]
"Multiple trajectory search for large scale global optimization"	(no)	(yes)	196	17.82	The authors proposed a new algorithm, named *multiple trajectory search (MTS)*. The basic concept of this algorithm was that many trajectories search the solution space and tree local search methods take place [82]
"The dynamic-Q optimization method: An alternative to SQP?"	(no)	(yes)	103	6.06	An algorithm with low storage requirements for handling constrained optimization problems has been proposed [73]
"A reassessment of the Snyman-Fatti dynamic search trajectory method for unconstrained global optimization"	(no)	(yes)	8	0.80	Snyman and Kok reassessment the well-known *multi-start global minimization algorithm with dynamic search trajectories* [71] and compare it with Storn and Price *differential evolution algorithm* [80] on an extensive set of problems [74]
"Global minimum cost design of a welded square stiffened plate supported at four corners"	(yes)	(no)	15	1.67	The authors have examined a well-known problem in the *steel structures* concerning the *stiffened plates* [20]

"TC" indicates the number of the total citations, while "CpY" indicates the number of citations per year (Source: Google Scholar)

Specifically, the Euler-Cauchy method (or explicit midpoint method) for solving the autonomous initial value problem (68) is given by:

$$y_{n+1} = y_n + h\, f\Big(y_n + \frac{1}{2}h\, f(y_n)\Big), \quad h = x_{n+1} - x_n, \tag{72}$$

and its corresponding Butcher's tableau is the following:

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

Furthermore, the Heun method is given by:

$$y_{n+1} = y_n + \frac{1}{2}h\, f(y_n) + \frac{1}{2}hf\Big(y_n + h\, f(y_n)\Big), \quad h = x_{n+1} - x_n. \tag{73}$$

and its corresponding Butcher's tableau is the following:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

Also, the Ralston method s given by:

$$y_{n+1} = y_n + \frac{1}{4}h\, f(y_n) + \frac{3}{4}hf\Big(y_n + \frac{2}{3}h\, f(y_n)\Big), \quad h = x_{n+1} - x_n. \tag{74}$$

and its corresponding Butcher's tableau is the following:

$$\begin{array}{c|cc} 0 & & \\ 2/3 & 2/3 & \\ \hline & 1/4 & 3/4 \end{array}$$

In general, for the second-order and two-stage autonomous Runge-Kutta methods the Butcher's tableau can be written as follows [13]:

$$\begin{array}{c|cc} 0 & & \\ c_2 & c_2 & \\ \hline & 1 - 1/(2c_2) & 1/(2c_2) \end{array}$$

and for any real finite value $c_2 > 0$ the corresponding methods have the form:

$$y_{n+1} = y_n + \left(1 - \frac{1}{2c_2}\right) h\, f(y_n) + \frac{1}{2c_2}hf\Big(y_n + c_2 h\, f(y_n)\Big), \quad h = x_{n+1} - x_n. \tag{75}$$

For example, using the above formula and the values $c_2 = 1/2$, $c_2 = 1$ and $c_2 = 2/3$ we obtain the methods (72), (73) and (74) correspondingly.

It is obvious that, if we apply the above approach to the initial value problem (70) we obtain the following family of an infinity number of methods for optimizing the objective function $F : \mathbb{R}^N \to \mathbb{R}$:

$$x^{k+1} = x^k - \left(1 - \frac{1}{2c_2}\right) h\, \nabla F\big(x^k\big) - \frac{1}{2c_2} h\, \nabla F\Big(x^k - c_2 h\, \nabla F\big(x^k\big)\Big), \quad k = 0, 1, \ldots, \tag{76}$$

Remark 4 It is obvious that an infinite number of optimization methods can be constructed since the corresponding number of Runge-Kutta methods is infinite.

Remark 5 Using the above approach it is easy to construct additional families of optimization methods using Runge-Kutta methods of various orders and stages.

The above methods have been successfully applied to several test functions and our experience is that the methods behaves predictably and reliably. The obtained results are comparable to those of Cauchy method (steepest descent method) and conjugate gradient methods. Let us point out here that, in general, the comparisons of different optimization methods is a hard issue. Also, due to the *no free lunch theorem for optimization* there is not a single algorithm that performs well on all problems and if an algorithm is improved for one particular problem, it will not perform well for other problems. Thus, on the average over all optimization problems, without re-sampling, all optimization algorithms perform equally well [89] (for a recent review see [1]). In addition, it is known that, for almost any pair of algorithms and various measures of algorithm performance such as execution time or solution quality, each algorithm will perform better than the other on some inputs [64].

Next, we present a strategy for developing *globally convergent algorithms* that is also applicable for the above proposed methods. The globally convergent algorithms have the property that the sequence of the iterates converge to a local minimizer of the objective function starting from almost any initial point [15].

Without loss of generality we rewrite the above proposed methods (76) to the following iterative scheme:

$$x^{k+1} = x^k + \alpha^k \varphi^k, \qquad k = 0, 1, \ldots, \tag{77}$$

where, at the kth iteration, $\alpha^k > 0$ determines the step size and $\varphi^k \neq 0$ the search direction.

The theoretical result presented below, allows us to equip the algorithms with a strategy for adapting the search direction to a descent one. Thus, a decrease of the function values at each iteration is ensured, and convergence to a local minimizer of the objective function is obtained from remote initial points.

Theorem 7 *Assume that: (a) the objective function $F : \mathbb{R}^n \to \mathbb{R}$ is bounded below in $\mathbb{R}^n$, (b) the gradient is Lipschitz continuous, i.e. there exists a constant $L > 0$ such that*

$$\|\nabla F(x) - \nabla F(y)\| \leqslant L\|x - y\|, \qquad \forall\, x, y \in \mathscr{N},$$

and (c) the function F is continuously differentiable in a neighborhood $\mathscr{N}$ of the level set $\mathscr{L} = \{x : F(x) \leqslant F(x^0)\}$, where x^0 is the starting point of the iterative scheme:

$$x^{k+1} = x^k + \alpha^k M^k \varphi^k, \qquad k = 0, 1, \ldots, \tag{78}$$

where $M^k = diag\{\mu_1^k, \mu_2^k, \ldots, \mu_n^k\}$ denotes the diagonal matrix with elements in $\mathbb{R}$, and $M^k\varphi^k \neq 0$. Suppose that the following relation is fulfilled:

$$\nabla F(x^k)^\top (M^k \varphi^k) < -\beta^k < 0, \tag{79}$$

and that $\alpha^k > 0$ satisfies the Wolfe's conditions:

$$F(x^k + \alpha^k M^k \varphi^k) - F(x^k) \leqslant \sigma_1 \alpha^k \nabla F(x^k)^\top (M^k \varphi^k), \tag{80}$$

$$\nabla F\left(x^k + \alpha^k (M^k \varphi^k)\right)^\top (M^k \varphi^k) \geqslant \sigma_2 \nabla F(x^k)^\top (M^k \varphi^k), \tag{81}$$

where $0 < \sigma_1 < \sigma_2 < 1$. Then the sequence $\{x^k\}_{k=0}^{\infty}$, generated by the iterative scheme (78) is globally convergent to a local minimizer of the objective function F.

Proof Following the proof of Theorem 6 of [86] due to Relation (79) the sequence $\{x^k\}_{k=0}^{\infty}$ of (78) follows a descent direction. Moreover, the Zoutendijk condition [96]:

$$\sum_{k\geqslant 1} \cos^2\theta_k \left\|\nabla F(x^k)\right\|^2 < \infty, \tag{82}$$

where

$$\cos\theta_k = \frac{-\nabla F(x^k)^\top (M^k \varphi^k)}{\|\nabla F(x^k)\| \, \|M^k \varphi^k\|}, \tag{83}$$

is fulfilled [90, 91, 96]. In our case Relation (83) becomes

$$\cos\theta_k = \frac{-\nabla F(x^k)^\top (M^k \varphi^k)}{\|\nabla F(x^k)\| \, \|M^k \varphi^k\|} > 0, \tag{84}$$

thus $\lim_{k\to\infty} \|\nabla f(x^k)\| = 0$, which means that the sequence of gradients converges to zero and the sequence $\{x^k\}_{k=0}^{\infty}$ is globally convergent to a local minimizer. Thus, the theorem is proved. □

Remark 6 Theorem 7 guarantees convergence to a local minimizer for any minimization algorithm that adopt the following strategy:

(a) define $(n-1)$, say $\{1, 2, \ldots, i-1, i+1, \ldots, n\}$, out of the n of the values $\{\mu_j^k\}_{j=1}^n$ of the elements of the matrix $M^k = \text{diag}\{\mu_1^k, \mu_2^k, \ldots, \mu_n^k\}$ and
(b) analytically calculate the remaining one μ_i^k in such a way that the Relation (79) to be fulfilled.

Note that no additional objective function or gradient evaluations are required since the proposed strategy uses pieces of information that are already be computed.

Remark 7 In various applications including neural network training and nonlinear least square problems among others, the objective function F is always bounded below, thus the Condition (a) of Theorem 7 is always fulfilled.

Although monotone convergence strategies provide an efficient and effective way to ensure that the error function is reduced sufficiently, they have the disadvantage that no information, which might accelerate convergence, is stored and used [21]. To alleviate this situation we can use a *nonmonotone convergence strategy* that exploits the accumulated information with regard to the most recent values of the objective function.

In [28] it has been shown that the condition which implies a monotonic decrease of $F(x^k)$ can be relaxed and yet global convergence can be established. To this end, the authors of [28] defined an acceptability criterion for the step size which can be viewed as a generalization of the Armijo rule [2] and they proved the following theorem:

Theorem 8 *Let* $\{x^k\}$ *be a sequence defined by*

$$x^{k+1} = x^k + \alpha^k d^k, \qquad d^k \neq 0.$$

Let $\alpha > 0$, $\sigma \in (0, 1)$, $\gamma \in (0, 1)$ *and let* W *be a nonnegative integer. Assume that:*

(i) the level set $\Omega_0 = \{x : F(x) \leqslant F(x^0)\}$ *is compact,*

(ii) *there exist positive numbers* q_1, q_2 *such that:*

$$\nabla F(x^k)^\top d^k \leqslant -q_1 \left\|\nabla F(x^k)\right\|^2, \tag{85}$$

$$\|d^k\| \leqslant q_2 \left\|\nabla F(x^k)\right\|, \tag{86}$$

(iii) $\alpha^k = \sigma^{h_k}\alpha$ *where* h_k *is the first nonnegative integer for which:*

$$F(x^k + \sigma^h \alpha\, d^k) \leqslant \max_{0\leqslant j\leqslant m(k)} \left\{F(x^{k-j})\right\} + \gamma\, \sigma^h \alpha\, \nabla F(x^k)\, d^k, \tag{87}$$

where $m(0) = 0$ *and* $0 \leqslant m(k) \leqslant \min\{m(k-1)+1, W\}$, $k \geqslant 1$.

Then:

(a) *the sequence* $\{x^k\}$ *remains in* Ω_0 *and every limit point* $\hat{x}$ *satisfies* $\nabla F(\hat{x}) = 0$,
(b) *no limit point of* $\{x^k\}$ *is a local maximizer of* F,
(c) *if the number of the stationary points o f* F *in* Ω_0 *is finite, the sequence* $\{x^k\}$ *converges.*

Remark 8 Condition (87) can be used to formulate a nonmonotone strategy that exploits the accumulated information with regard to the most recent values of the objective function. Also, it defines a criterion of acceptance of any iterate using a nonnegative integer W, named *nonmonotone window* or *nonmonotone horizon*. This condition allows for an increase in the function values, which is regulated by the value of γ, without affecting the global convergence properties [28, 63]. In practice, a value of $0 < \gamma \ll 1$ is suggested. Finally, the nonmonotone strategy has been efficiently applied in the case of training of multilayer feedforward neural networks that can be considered as a highly nonlinear minimization problem, involving sigmoid functions that have infinitely broad regions with arbitrary small derivatives [56–58].

Remark 9 It follows directly from Theorem 8 that the nonmonotone strategy generates a globally convergent sequence for any algorithm that follows search direction d^k, provided that the two positive numbers q_1, q_2 exist such that the Relation (85) and the Relation (86) are fulfilled. Obviously, the same holds in the case of the search direction $M^k \varphi^k$ of Theorem 7.

5 Synopsis and future research work

A detailed survey of traditional and widely used dynamic search trajectory methods for global optimization is given. Also, the most recent of these methods and their applications are presented. In addition, in order to inform the reader about the applicability and the interest gathered regarding these methods a citation analysis using the most well-known and most cited methods is provided.

Furthermore, families of an infinity number of dynamic search trajectories methods for global optimization based on the numerical solution of autonomous ordinary differential equations are given. Also, a strategy for developing globally convergent methods that is applicable to the proposed families of methods is given and the corresponding convergence theorem is proved. Finally, theoretical results for obtaining nonmonotone convergent methods that exploit the accumulated information with regard to the most recent values of the objective function are given. In a future correspondence, comparative numerical results by

considering the above methods will be given and their performance will be studied comprehensively. Furthermore, we will study their performance under the influence of globally convergent and nonmonotone convergent strategies.

Finally, in a future correspondence we intend to apply our approach on second order ordinary differential equation. In addition, besides the second order Runge-Kutta methods that are presented in this paper various other Runge-Kutta methods for the numerical solution of initial value problems will be investigated and the corresponding dynamic search trajectory methods for global optimization will be presented and will be analyzed.

Acknowledgements The authors wish to thank the three anonymous reviewers for their helpful comments. S.-A. N. Alexandropoulos is supported by Greece and the European Union (European Social Fund-ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning" in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY). P. M. Pardalos is supported by the Paul and Heidi Brown Preeminent Professorship at ISE (University of Florida, USA), and a Humboldt Research Award (Germany).

References

1. Adam, S.P., Alexandropoulos, S.-A.N., Pardalos, P.M., Vrahatis, M.N.: No free lunch theorem: A review. Approximation and optimization. In: Demetriou, I.C., Pardalos, P.M. (eds.) Springer Optimization and Its Applications, vol. 145, pp. 57–82. Springer International Publishing AG, Cham (2019)
2. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**(1), 1–3 (1966)
3. Askarzadeh, A.: A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Comput. Struct. **169**, 1–12 (2016)
4. Back, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies,Evolutionary Programming, Genetic Algorithms. Oxford University Press (1996)
5. Barbieri, C., Cocco, S., Monasson, R.: On the trajectories and performance of infotaxis, an information-based greedy search algorithm. EPL (Europhysics Letters) **94**(2), 20005-p1–p6 (2011)
6. Battiti, R.: First-and second-order methods for learning: Between steepest descent and Newton's method. Neur. Comput. **4**(2), 141–166 (1992)
7. Bhaya, A., Kaszkurewicz, E.: Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method. Neural Netw. **17**(1), 65–71 (2004)
8. Bhaya, A., Kaszkurewicz, E.: Control Perspectives on Numerical Algorithms and Matrix Problems, vol. 10. SIAM (2006)
9. Bhaya, A., Pazos, F., Kaszkurewicz, E.: The controlled conjugate gradient type trajectory-following neural net for minimization of nonconvex functions. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2010)
10. Boggs, P.T.: An algorithm, based on singular perturbation theory, for ill-conditioned minimization problems. SIAM J. Numer. Anal. **14**(5), 830–843 (1977)
11. Branin, F.H.: Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. IBM J. Res. Dev. **16**(5), 504–522 (1972)
12. Butcher, J.C.: The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. Wiley-Interscience (1987)
13. Butcher, J.C.: Numerical Analysis of Ordinary Differential Equations, 2nd edn. Wiley (2008)
14. Cesari, L.: Optimization Theory and Applications: Problems with Ordinary Differential Equations, vol. 17. Springer Science & Business Media (2012)
15. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Classics in Applied Mathematics. SIAM (1996)
16. Dixon, L.C.W.: Neural networks and unconstrained optimization. In: Algorithms for Continuous Optimization, pp. 513–530. Springer (1994)
17. Dixon, L.C.W., Szego, G.P.: The global optimization problem: An introduction. Toward Global Optim. **2**, 1–15 (1978)
18. D'yakonov, E.G.: Optimization in Solving Elliptic Problems. CRC Press (2018)

19. Farkas, J., Jármai, K.: Design and Optimization of Metal Structures. Elsevier (2008)
20. Farkas, J., Jármai, K., Snyman, J.A.: Global minimum cost design of a welded square stiffened plate supported at four corners. Struct. Multidiscip. Optim. **40**(1–6), 477 (2010)
21. Fiacco, A.V., McCormick, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Classics in Applied Mathematics. SIAM (1990)
22. Fletcher, R.: Fortran subroutines for minimization by quasi-Newton methods. Technical Report AERE-R–7125. Atomic Energy Research Establishment Harwell, England (1972)
23. Fletcher, R.: Practical Methods of Optimization. Wiley (2013)
24. Floudas, C.A., Pardalos, P.M.: A Collection of Test Problems for Constrained Global Optimization Algorithms, pp. 455. Springer Science & Business Media (1990)
25. Gill, P.E., Murray, W.: Quasi-Newton methods for unconstrained optimization. IMA J. Appl. Math. **9**(1), 91–108 (1972)
26. Griewank, A.O.: A Generalized Descent Method for Global Optimization. Master Thesis The Australian National University (1977)
27. Griewank, A.O.: Generalized descent for global optimization. J. Optim. Theory Appl. **34**(1), 11–39 (1981)
28. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**(4), 707–716 (1986)
29. Groenwold, A.A., Snyman, J.A.: Global optimization using dynamic search trajectories. Combinatorial and global optimization. In: Pardalos, P.M., Migdalas, A., Burkard, R. (eds.) Series on Applied Mathematics, vol. 14, pp. 123–132. World Scientific Publishing Co (2002)
30. Groenwold, A.A., Snyman, J.A., Stander, N.: Modified trajectory method for practical global optimization problems. AIAA J. **34**(10), 2126–2131 (1996)
31. Hairer, E., Nørsett, S.P., Wanner, G.: Solving ordinary differential equations I: Nonstiff problems. Springer Series in Computational Mathematics, p. 1. Springer (2009)
32. Henrici, P.: Discrete Variable Methods in Ordinary Differential Equations. Wiley (1962)
33. Horst, P., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization, 2nd edn. Kluwer Academic Publishers (2000)
34. Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill Companies (1972)
35. Incerti, S., Parisi, V., Zirilli, F.: A new method for solving nonlinear simultaneous equations. SIAM J. Numer. Anal. **16**(5), 779–789 (1979)
36. Inomata, S., Cumada, M.: On the golf method. Bull. Electron. Laboratory **25**(3), 495–512 (1964)
37. Jain, M.K.: Numerical Solution of Differential Equations. Wiley, Eastern New Delhi (1979)
38. Kam, T.Y., Chang, R.R.: Optimal design of laminated composite plates with dynamic and static considerations. Comput. Struct. **32**(2), 387–393 (1989)
39. Kam, T.Y., Lai, M.D.: Multilevel optimal design of laminated composite plate structures. Comput. Struct. **31**(2), 197–202 (1989)
40. Kam, T.Y., Snyman, J.A.: Optimal design of laminated composite plates using a global optimization technique. Compos. Struct. **19**(4), 351–370 (1991)
41. Kan, A.R., Timmer, G.T.: Stochastic methods for global optimization. Am. J. Math. Manag. Sci. **4**(1-2), 7–40 (1984)
42. Kazarlis, S.A., Papadakis, S.E., Theocharis, J., Petridis, V.: Microgenetic algorithms as generalized hill-climbing operators for GA optimization. IEEE Trans. Evol. Comput. **5**(3), 204–217 (2001)
43. Lambert, J.D.: Numerical Methods for Ordinary Differential Systems: The Initial Value Problem. Wiley (1991)
44. Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Evolutionary operators in global optimization with dynamic search trajectories. Numer. Algor. **34**(2–4), 393–403 (2003)
45. Leung, Y.W., Wang, Y.: An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans. Evol. Comput. **5**(1), 41–53 (2001)
46. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009)
47. Massard, T.N.: Computer sizing of composite laminates for strength. J. Reinforced Plastics Compos. **3**(4), 300–345 (1984)
48. Migdalas, A., Pardalos, P.M.: A note on open problems and challenges in optimization theory and algorithms. In: Open Problems in Optimization and Data Analysis, vol. 141, pp. 1–8. Springer International Publishing AG, Cham (2018)
49. Pardalos, P.M., Migdalas, A. (eds.): Open Problems in Optimization and Data Analysis, vol. 141. Springer International Publishing AG, Cham (2018)
50. Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization method for constrained optimization problems. Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies **76**(1), 214–220 (2002)

51. Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization method in multiobjective problems. In: Proceedings of the ACM Symposium on Applied Computing (SAC 2002), pp. 603–607 (2002)
52. Parsopoulos, K.E., Vrahatis, M.N.: Particle Swarm Optimization and Intelligence: Advances and Applications. Information Science Publishing, IGI Global (2010)
53. Petalas, Y.G., Tasoulis, D.K., Vrahatis, M.N.: Trajectory methods for neural network training. In: Hamza, M.H. (ed.) Artificial Intelligence and Applications, vol. 1, pp. 400–408. IASTED/ACTA Press, USA (2004)
54. Petalas, Y.G., Tasoulis, D.K., Vrahatis, M.N.: Dynamic search trajectory methods for neural network training. Lect. Notes Comput. Sci. (LNAI) **3070**, 241–246 (2004)
55. Petalas, Y.G., Vrahatis, M.N.: Trajectory methods for supervised learning. In: Proceedings of the First International Conference from Scientific Computing to Computational Engineering (IC-SCCE 2004), September 8-10, Athens (2004)
56. Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Nonmonotone learning rules for backpropagation networks. In:: Proceedings of the Sixth IEEE International Conference on Electronics, Circuits and Systems (ICECS 1999), September 5-8, 1999, Pafos, Cyprus, vol. 1, art.no. 812280, pp. 291–294. IEEE (1999)
57. Plagianakos, V.P., Magoulas, G.D., Vrahatis, M.N.: Deterministic nonmonotone strategies for effective training of multilayer perceptrons. IEEE Trans. Neural Netw. **13**(6), 1268–1284 (2002)
58. Plagianakos, V.P., Vrahatis, M.N., Magoulas, G.D.: Nonmonotone methods for backpropagation training with adaptive learning rate. In: IEEE Proceedings of the International Joint Conference on Neural Networks (IJCNN 1999), July 10-16, 1999, Washington DC, USA, vol. 3, art.no. 832644, pp. 1762–1767, IEEE (1999)
59. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)
60. Powell, M.J.: A fast algorithm for nonlinearly constrained optimization calculations. In: Numerical Analysis, pp. 144–157. Springer (1978)
61. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C. Cambridge University Press (2002)
62. Price, W.: Global optimization by controlled random search. J. Optim. Theory Appl. **40**(3), 333–348 (1983)
63. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**(1), 26–33 (1997)
64. Roughgarden, T.: Beyond worst-case analysis. Commun. ACM **62**(3), 88–96 (2019)
65. Salvadori, L.: Famiglie ad un parametro di funzioni di Liapunov nello studio della stabilita. In: Symposia Mathematica, vol. 6, pp. 309–330. Academic Press (1971)
66. Schäffler, S., Warsitz, H.: A trajectory-following method for unconstrained optimization. J. Optim. Theory Appl. **67**(1), 133–140 (1990)
67. Sengupta, S., Basak, S., Peters, R.: Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives. Mach. Learn. Knowl. Extract. **1**(1), 157–191 (2018)
68. Shang, Y., Wah, B.W.: Global optimization for neural network training. Computer **29**(3), 45–54 (1996)
69. Snyman, J.A.: A new and dynamic method for unconstrained minimization. Appl. Math. Model. **6**(6), 449–462 (1982)
70. Snyman, J.A.: An improved version of the original leap-frog dynamic method for unconstrained minimization: LFOP1(b). Appl. Math. Model. **7**(3), 216–218 (1983)
71. Snyman, J.A., Fatti, L.P.: A multi-start global minimization algorithm with dynamic search trajectories. J. Optim. Theory Appl. **54**(1), 121–141 (1987)
72. Snyman, J.A., Geerthsen, K.A.: The practical application of a dynamic search-trajectory method for constrained global optimization. In: IUTAM Symposium on Optimization of Mechanical Systems, pp. 285–292. Springer (1996)
73. Snyman, J.A., Hay, A.M.: The dynamic-q optimization method: An alternative to SQP? Comput. Math. Appl. **44**(12), 1589–1598 (2002)
74. Snyman, J.A., Kok, S.: A reassessment of the Snyman-Fatti dynamic search trajectory method for unconstrained global optimization. J. Glob. Optim. **43**(1), 67–82 (2009)
75. Snyman, J.A., Stander, N., Roux, W.J.: A dynamic penalty function method for the solution of structural optimization problems. Appl. Math. Model. **18**(8), 453–460 (1994)
76. Snyman, J.A., Wilke, D.N.: New Gradient-Based Trajectory and Approximation Methods. Springer Optimization and its Applications. In: Pardalos, P.M. (ed.), vol. 133, pp. 197–250. Springer International Publishing AG, Cham (2018)
77. Snyman, J.A., Wilke, D.N.: Practical mathematical optimization: Basic optimization theory and gradient-based algorithms. In: Pardalos, P.M. (ed.) Springer Optimization and its Applications. 2nd edn., vol. 133. Springer International Publishing AG, Cham (2018)

78. Soliman, S.A.H., Mantawy, A.A.H.: Modern Optimization Techniques with Applications in Electric Power Systems. Springer Science & Business Media (2011)
79. Sörensen, K.: Metaheuristics-the metaphor exposed. Int. Trans. Oper. Res. **22**(1), 3–18 (2015)
80. Storn, R., Price, K.: Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
81. Tseng, L.Y., Chen, C.: Multiple trajectory search for multiobjective optimization. In: IEEE Congress on Evolutionary Computation, (CEC 2007), pp. 3609–3616. IEEE (2007)
82. Tseng, L.Y., Chen, C.: Multiple trajectory search for large scale global optimization. In: IEEE Congress on Evolutionary Computation, (CEC 2008), (IEEE World Congress on Computational Intelligence), pp. 3052–3059. IEEE (2008)
83. Vincent, T., Goh, B., Teo, K.: Trajectory-following algorithms for min-max optimization problems. J. Optim. Theory Appl. **75**(3), 501–519 (1992)
84. Vogl, T.P., Mangis, J., Rigler, A., Zink, W., Alkon, D.: Accelerating the convergence of the back-propagation method. Biol. Cybern. **59**(4–5), 257–263 (1988)
85. Vavasis, S.A.: Complexity issues in global optimization: A survey. In: Horst, R., Pardalos, P.M. (eds.) Handbook of Global Optimization, pp. 27–41. Kluwer Academic (1995)
86. Vrahatis, M.N., Magoulas, G.D., Plagianakos, V.P.: From linear to nonlinear iterative methods. Appl. Numer. Math. **45**(1), 59–77 (2003)
87. Wales, D.J., Doye, J.P.: Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. J. Phys. Chem. A **101**(28), 5111–5116 (1997)
88. Walter, W.: Gewöhnliche differentialgleichungen. Springer (2000)
89. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
90. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**, 226–235 (1969)
91. Wolfe, P.: Convergence conditions for ascent methods II: Some corrections. SIAM Rev. **13**, 185–188 (1971)
92. Yi, J.H., Deb, S., Dong, J., Alavi, A.H., Wang, G.G.: An improved nsga-iii algorithm with adaptive mutation operator for big data optimization problems. Future Generation Computer Systems (2018)
93. Zabinsky, Z.B.: Stochastic Adaptive Search for Global Optimization. Springer (2003)
94. Zidkov, N., Siedrin, B.: A certain method of search for the minimum of a function of several variables. Comput. Methods Program. **10**, 203–210 (1968)
95. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
96. Zoutendijk, G.: Nonlinear Programming, Computational Methods. In: Abadie, J. (ed.) Integer and Nonlinear Programming, pp. 37–86, North–Holland (1970)