MATIMYÁS MATEMATIKA
ISSN 0115-6926

Journal of the Mathematical Society of the Philippines
Vol. 48 No. 2 (2025) pp. 9-29

Accelerated gradient descent algorithm with Golden Section search for escaping saddle points in optimization problems

Rey Audie S. Escosio
Institute of Mathematics
University of the Philippines Diliman
Quezon City, Philippines

Renier G. Mendoza
Institute of Mathematics
University of the Philippines Diliman
Quezon City, Philippines
rmendoza@math.upd.edu.ph

Abstract

Gradient-based algorithms rely on derivative information to determine descent directions and are known for their fast convergence and low computational cost compared to metaheuristic methods. However, despite extensive developments addressing issues such as saddle-point avoidance and local trapping, many existing approaches either require higher-order information, stochastic perturbations, or problem-specific heuristics, and often lose efficiency or robustness near stationary points.

This work addresses the gap between fast gradient-based methods and reliable local exploration by proposing a hybrid deterministic algorithm, termed AGD–nGSS, which couples Nesterov's accelerated gradient descent (AGD) with an n-dimensional golden section search (nGSS). In the proposed framework, AGD is employed for efficient global descent until a stationary point is detected, after which a localized, derivative-free nGSS is activated to verify optimality and potentially escape saddle points or local minimizers.

The effectiveness of the proposed method is evaluated in terms of convergence accuracy, robustness with respect to initialization, and consistency in attaining the global minimum. Numerical experiments on standard benchmark functions compare AGD–nGSS against classical gradient-based methods, showing improved global convergence behavior across a wide range of initial points while maintaining competitive computational efficiency. Applications to matrix factorization and parameter estimation further demonstrate the practical relevance of the approach.

Keywords: gradient descent methods, accelerated gradient descent, golden section search, matrix decomposition, parameter estimation

2020 MSC: 65K05, 90C26, 90C30

1 Introduction

The method of gradient descent (GD) was formulated by Cauchy in 1847 [1] and has been a staple of gradient-based optimization. This approach implements the negative gradient as its descent direction, i.e., $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$. With a positive multiplier η, the factor $-\eta\nabla f(\mathbf{x}_k)$ is deemed to be the maximum rate of descent until convergence [2]. Thus, the succession of GD is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\nabla f(\mathbf{x}_k), \tag{1}$$

where $\mathbf{x}_k$ denotes the current iterate and $\mathbf{x}_{k+1}$ the subsequent iterate at iteration k. The iteration is terminated when one of the following stopping criteria is satisfied:

- the maximum number of iterations is reached, i.e., $k \geq k_{\max}$;
- the gradient norm is sufficiently small, i.e.,

$$\|\nabla f(\mathbf{x}_k)\| \leq \varepsilon;$$

- the relative change between successive iterates is below a tolerance,

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}{\|\mathbf{x}_k\|} \leq \varepsilon.$$

The convergence rate of GD is only of the order $\mathcal{O}(1/k)$ for a convex problem with Lipschitz-continuous gradient. This is the case for both constant and adaptive step sizes. Additionally, global convergence is not guaranteed. The algorithm may alternatively arrive at a local minimum or a saddle point since these may also satisfy the termination condition dependent on the gradient. Standalone GD cannot converge properly in problems containing steep slopes, multiple local minima, null spaces, and saddle points that collectively hinder global convergence.

A standout series of papers started by Nesterov [3] found a possible increase of the convergence rate of the gradient method to $\mathcal{O}(1/k^2)$ via acceleration. By looking at the lower complexity bounds of differentiable and L-smooth convex functions [4], the study pointed out that a sublinear rate of convergence is attainable for first-order methods. This finding led to a modification of GD that optimizes at a sublinear rate of $\mathcal{O}(1/k^2)$ called accelerated gradient descent (AGD).

The method enhances the optimization with the use of "momentum" which is the iteration

$$\mathbf{y}_k = \mathbf{x}_k + (1-\theta)\mathbf{v}_k, \tag{2}$$

where $\mathbf{x}_k$ is the current step, $\mathbf{v}_{k+1} = \mathbf{x}_{k+1} - \mathbf{x}_k$, and $\mathbf{y}_k$ is the inferred step. Equation (2) would then factor in the previous steps in the term $\mathbf{v}_k$ with a limiting constant θ that we will call the momentum coefficient. The obtained $\mathbf{y}_k$ is then used to determine the next

step $\mathbf{x}_{k+1}$, i.e.,

$$\mathbf{x}_{k+1} = \mathbf{y}_k - \eta_k \nabla f(\mathbf{y}_k) \tag{3}$$

and this proceeds iteratively until it converges to a stationary point. Noticeably, the method is reduced to a mere GD if $\theta = 1$ since it eliminates completely the term $\mathbf{v}_k$.

The AGD updates with a factor from the gradient history. This included acceleration helps with the convergence by limiting the computational cost. However, the method may still encounter hindrances to converge globally due to unwanted stationary points and the nature of the objective function.

Some modified works on GD [5] and AGD [6] perturb the iteration with a random variable sampled at a neighborhood centered at the stationary point. As a result, these algorithms escape the saddle points. Others have created methods by alternating between gradient and Hessian computations in a stochastic GD [7], by looking for higher-order derivatives in other second-order algorithms [8], and by resetting stochastic GD on certain iteration conditions [9]. These variants have shown capabilities to improve the convergence of gradient-based methods. However, these methods are designed as local minimizers.

Similarly, some studies combine known methods to create hybrid algorithms. These are methods that validate the determined optimum of their main algorithm by running independently or co-dependently. Studied techniques are created as combinations of particle swarm and GD [10], genetic algorithm and conjugate gradient (CG) [11], differential evolution and Broyden–Fletcher–Goldfarb–Shanno (BFGS) [12], stochastic GD and dual coordinate descent [13], chaotic maps and golden section search [14], and augmented Lagrangian algorithm and multiobjective GD [15]. Although capable of global convergence, these algorithms require high computational time. They are also probabilistic and so a result of one run may be different from another.

Despite the efficiency of accelerated gradient-based methods, their performance may suffer near stationary points, particularly in nonconvex landscapes where saddle points and local minimizers are prevalent. Existing methods often rely on higher-order derivatives, stochastic perturbations, or problem-specific heuristics, which can increase computational cost or reduce robustness. There remains a need for a deterministic optimization framework that preserves the fast convergence of gradient-based methods while incorporating a principled local mechanism to validate and refine stationary points.

This paper aims to develop a gradient-based algorithm augmented with a secondary local search. Specifically, we combine Nesterov's accelerated gradient descent (AGD) with the n-dimensional golden section search (nGSS). A theoretical convergence result for the nGSS component is also provided.

We test this hybridized technique on unimodal and multimodal benchmark functions, matrix decomposition, and parameter estimation.

The advantages of the proposed method are as follows;

- *deterministic:* the same initial guess will result in the same final estimate,
- *escapes saddle point:* convergence of nGSS does not rely on the gradient of f, and

- *global convergence:* capable of global convergence by applying partitioning to nGSS.

In the next section, we discuss in detail the formulation of the main algorithm. Then, we present our numerical results and compare them with other standard approaches. We also present applications of the proposed method. Finally, we end with some concluding remarks and future works.

2 Methods

2.1 Golden Section Search, its Generalization in Higher Dimensions, and its Convergence

The golden section search (GSS) is a single variable search method used to determine the minimum inside an interval. The technique, and the title itself, is based on the golden ratio φ via a derived quantity ϕ, i.e.,

$$\phi = \frac{\sqrt{5}-1}{2} = 1 - \varphi \approx 0.61803.$$

It uses this quantity to sequentially narrow the interval and compare the function values of selected points until it converges towards a minimum. In essence, GSS is optimal for unimodal functions and does not need the objective function to be continuous [16]. For multimodal functions, GSS will iterate towards a minimizer that may or may not be global.

The iterative scheme of GSS begins with an interval $[a, b]$ in the domain such that $a < x^* < b$, where x^* is the minimizer of f. The first step is to determine two arbitrary points x_1 and x_2 inside the domain, i.e.,

$$x_1 = a + (1-\phi)(b-a) \text{ and } x_2 = a + \phi(b-a). \tag{4}$$

The use of coefficients $(1-\phi)$ and ϕ ensures that these points lie within the interval.

The points x_1 and x_2 are compared with their function values. Without loss of generality, suppose $f(x_1) < f(x_2)$. Then, the next iteration reassigns the boundary to the point with the higher function value such that the next interval is $[a, x_2]$. The process continues until the termination condition converges to a point, taken to be the minimum. The proof of convergence of nGSS for unimodal functions is presented in the next section.

GSS has been used in different applications such as object tracking [17], image processing [18], and signal equalization [19]. Further research have remodeled GSS for higher dimensions that follow the structure of its 1-d variant. This generalization has been discussed in [20, 21].

The extension to higher dimensions of GSS is called the n-dimensional golden section search (nGSS). Like GSS, this method in dimension n requires an initial search space $\Omega_0^{(n)}$ generated by the points $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ containing the least and greatest elements of the boundaries of each dimension, respectively. It should be noted that we assume $\mathbf{x}^*$ exists inside

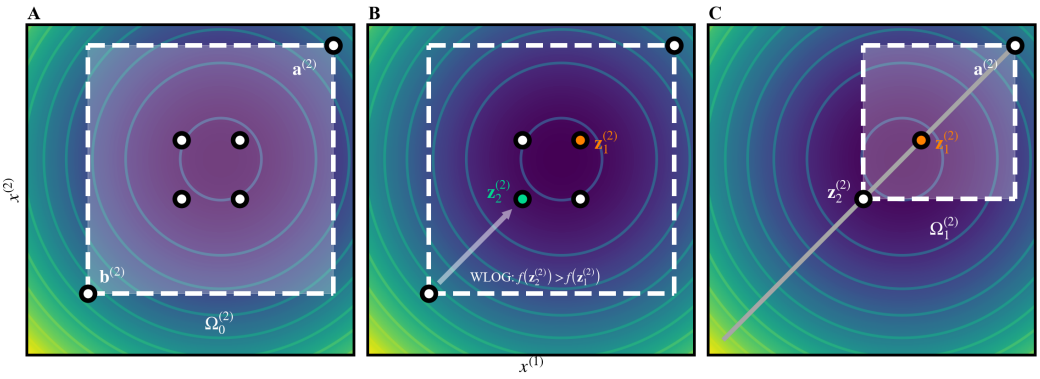

Figure 1: **An example minimization of two-dimensional GSS**. (A) The algorithm initializes at a search space $\Omega_0^{(2)}$ of a unimodal function. (B) Four arbitrary points are determined but we highlight $z_1^{(2)}$ (orange) and $z_2^{(2)}$ (green). The orange point has the least function value so it is retained. (C) The green point will be the new bound with the search space reduced to $\Omega_1^{(2)}$. Create a plane (or hyperplane for n dimensional cases, shown as the gray line) that includes the points $x_1^{(2)}$ and $x_2^{(2)}$.

$\Omega_0^{(n)}$, i.e., $a_j < x_j^* < b_j$, where the elements are the jth element of $\mathbf{a}$, $\mathbf{x}^*$, and $\mathbf{b}$, respectively, and j runs from 1 to n.

For each dimension j, two arbitrary points are determined, i.e.,

$$x_{1,j} = a_j + (1-\phi)(b_j - a_j) \text{ and } x_{2,j} = a_j + \phi(b_j - a_j), \tag{5}$$

to get 2^n number of points that lie inside $\Omega_0^{(n)}$. These are all collected inside the Cartesian product $\mathbf{x}_1 \times \mathbf{x}_2$ wherein $\mathbf{x}_1$ and $\mathbf{x}_2$ contain all $x_{1,j}$ and $x_{2,j}$ for all j, respectively.

We let $\mathbf{m}$ be the point in $\mathbf{x}_1 \times \mathbf{x}_2$ that induces the least function value among all gathered points, i.e.,

$$\mathbf{m} = \underset{\mathbf{x} \in \mathbf{x}_1 \times \mathbf{x}_2}{\arg\min} f(\mathbf{x}).$$

Then, $\mathbf{m}$ remains on the succeeding search space $\Omega_1^{(n)}$ while the point opposite to it (or the point with the exactly different value for all dimensions) replaces its nearest boundary of the search space. The succeeding search space $\Omega_1^{(n)}$ is a subset of the original search space i.e., $\Omega_1^{(n)} \subset \Omega_0^{(n)}$. Additionally, the minimizer $\mathbf{x}^*$ of the function must be well-contained as the n-dimensional volume decreases in size until it terminates. The method outputs a guaranteed minimizer at a specified tolerance. This is shown in Figure 1 A-C. Note that in this process, the endpoints of the decreasing-in-size search space can change for all dimensions j.

2.2 Modifications to the Local Search

The introduced local search assists the gradient-based algorithm to resolve some of its limitations, but creating it may still encounter complications on convergence. We explore the use of a generalization of the spherical coordinate system and partitioning in improving the search process of the nGSS.

The n-dimensional spherical coordinate system (nSCS) uses the n-sphere defined as $S^n = \left\{x \in \mathbb{R}^{n+1} \ : \ ||x|| = \rho\right\}$, where $\rho \in \mathbb{R}$ and $||\cdot||$ is the Euclidean metric. This considers a set of points in a given dimension n embedded in an $n+1$ Cartesian space.

The nSCS reduces the parameters of nGSS to only one: the maximum radius length ρ of the sphere. Additionally, it creates equidistant boundaries around the iterate.

Partitioning the search space can also be done. It works by subdividing the domains of each dimension into smaller, equal intervals using a partition number p. In each region, the local search selects its candidate as the minimizer in that search space. Out of all candidates, the point with the smallest function value is considered the minimizer.

2.3 The Main Algorithm

Algorithm 1 Accelerated gradient descent with the n-dimensional golden section search

Input: function f, initial guess $\mathbf{x}_0$, maximum iteration N, gradient tolerance $\varepsilon_{\text{grad}}$, numerical tolerance ε_{num}, momentum coefficient θ, maximum radius length ρ, number of partitions p

Output: minimizer $\mathbf{x}^*_{k+1}$

$\mathbf{v}_0 = 0$
for $k = 0, 1, 2, \ldots, N$ **do**
  $\eta_k \leftarrow \arg\min_{\eta} f(\mathbf{x}_k - \eta \nabla f(\mathbf{x}_k))$
  $\mathbf{y}_k \leftarrow \mathbf{x}_k + (1-\theta)\mathbf{v}_k$
  $\mathbf{x}_{k+1} \leftarrow \mathbf{y}_k - \eta_k \nabla f(\mathbf{y}_k)$
  $\mathbf{v}_{k+1} \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k$
  if $||\nabla f(\mathbf{x}_{k+1})|| < \varepsilon_{\text{grad}}$ **then**
    $\mathbf{x}^*_{k+1} \leftarrow \arg\min_{\mathbf{x} \in \Omega_\rho(\mathbf{x}_{k+1})} f(\mathbf{x})$
    if $||\mathbf{x}^*_{k+1} - \mathbf{x}_{k+1}|| < \varepsilon_{\text{num}}$ **then**
      break
    else
      $\mathbf{x}_{k+1} \leftarrow \mathbf{x}^*_{k+1}$
    end if
  end if
end for

Global search (AGD)

Local search (nGSS)

By combining AGD and the local search (nGSS with nSCS and partitioning), the algorithm can be called the accelerated gradient descent with the n-dimensional golden section search, abbreviated as AGD-nGSS and is shown in Algorithm 1.

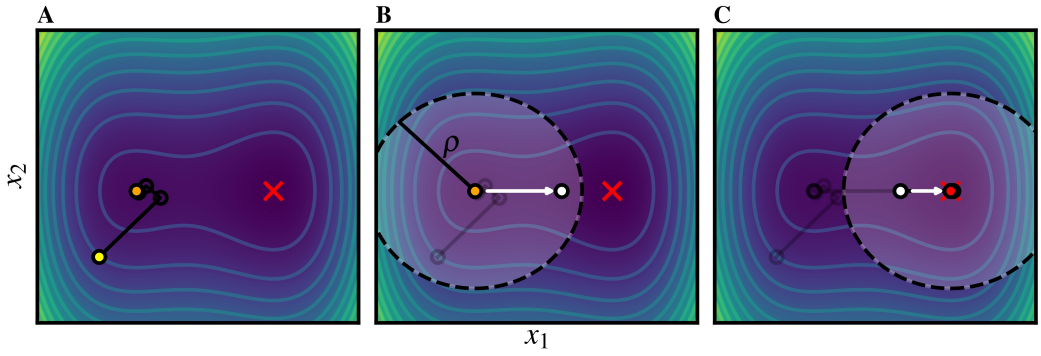

Figure 2: **Iterations of AGD-nGSS passing through a stationary point**. We want the algorithm to determine the global minimum (red cross). (A) From an initial point (yellow filled), the iteration encounters a stationary point (orange filled) via AGD. (B) It then triggers the modified local search with the predefined radius ρ, as annotated, of the search space (black, dashed). A minimizer (white filled) of the search space is obtained and immediately iterated upon (white arrow). (C) The algorithm returns to AGD from that point and encounters yet again another stationary point (red filled). Upon inspection, the determined minimizer is still at the same iterate. Hence, the method converges with the optimizer.

In the algorithm, Lines 1 to 6 describe the adaptive AGD as it passes through the function to determine stationary points. Once it encounters a stationary point by triggering the termination condition in Line 7, the modified local search will be activated in Line 8. A localized neighborhood Ω_ρ from this iterate is created and probed with its own minimizer via the modified local search.

The determined minimizer from the local search validates the stationary point that AGD came across, seen in Lines 9 to 13. This is done by comparing the norm of the difference between the iterate and the determined minimizer from the modified local search to be within a certain tolerance. A general sense of convergence for stationary points can be seen in Figure 2.

The method naturally requires the objective function f and the initial guess $\mathbf{x}_0$. The step length η, while considered a parameter, is adaptively determined using 1GSS, an exact line search method. Other parameters and their chosen values will be defined in the next section.

2.4 Benchmark Tests

In optimization, benchmark functions embody the surfaces and structures that an algorithm encounters in mathematical problems. They may vary on the depth and shape of the neighborhood, dimensions of the problem, and the multimodality and nonconvexity of the functions [22]. In this research, we considered only differentiable functions since AGD is gradient-based. The benchmark functions used in this study are from [23].

Table 1: Chosen parameter values of AGD-nGSS for the numerical experiments.

Parameter	**Variable**	**Value**
maximum iteration	N	1500
gradient tolerance	$\varepsilon_{\text{grad}}$	10^{-4}
numerical tolerance	ε_{num}	10^{-7}
momentum coefficient	θ	0.95
maximum radius length	ρ	$\frac{1}{3}\min\{\mathbf{x}_{\text{upper}} - \mathbf{x}_{\text{lower}}\}$
number of partitions	p	4

The main quantity used to indicate global convergence is the function value of the minimizer. We say that a certain optimization run converged globally if

$$f_{\text{opt}} - f(\mathbf{x}^*) < \varepsilon_{\text{fval}}, \tag{6}$$

where f_{opt} is the function value of the determined minimizer and $f(\mathbf{x}^*)$ is the global minimum, and $\varepsilon_{\text{fval}}$ is a chosen tolerance value.

The parameters used for AGD-nGSS are provided in Table 1, corresponding to the requirements of Algorithm 1. For the maximum radius length, the vectors $\mathbf{x}_{\text{upper}}$ and $\mathbf{x}_{\text{lower}}$ are the upper and lower bounds of the problem, respectively. Multiplying the constant $\frac{1}{3}$ helps maintain the neighborhood size of the local search for any problem.

We also test the main algorithm in solving two real-world problems:

1. *matrix factorization:* which has many applications in data science [24, 25, 25, 26], and
2. *parameter estimation:* which is a key technique to fit data in a mathematical model [27, 28, 29].

3 Results and Discussions

3.1 Convergence of nGSS

The following theorem explains how nGSS converges for a unimodal function.

Theorem 1. *Let $f : \Omega_0^{(n)} \to \mathbb{R}$ be strictly convex on $\Omega_0^{(n)} = [\mathbf{a}_0, \mathbf{b}_0] \subset \mathbb{R}^n$, and let $\mathbf{x}^*$ be its unique global minimizer. Assume further that f satisfies the following corner-monotonicity property:*

(Corner-monotonicity) *For any two points $\mathbf{x}, \mathbf{y} \in \Omega_0^{(n)}$, if*

$$|x_j - x_j^*| \leq |y_j - x_j^*| \quad \text{for all } j = 1, \ldots, n,$$

with strict inequality for at least one index, then $f(\mathbf{x}) < f(\mathbf{y})$.

Let $\{\Omega_k^{(n)}\}_{k\geq 0}$ be the sequence of search spaces generated by the n-dimensional golden section search (nGSS) algorithm defined by Equation (5)*. Then:*

- $\Omega_{k+1}^{(n)} \subseteq \Omega_k^{(n)}$ *for all* k*;*
- $\mathbf{x}^* \in \Omega_k^{(n)}$ *for all* k*;*
- $\mathbf{a}_k^{(n)} \to \mathbf{x}^*$ *and* $\mathbf{b}_k^{(n)} \to \mathbf{x}^*$ *as* $k \to \infty$.

Proof. We show nestedness, invariance of the minimizer, and geometric shrinkage.

Nestedness. At iteration k, for each coordinate $j = 1, \ldots, n$, the algorithm constructs the interior points

$$x_{1,j} = a_k^{(j)} + (1-\phi)(b_k^{(j)} - a_k^{(j)}), \qquad x_{2,j} = a_k^{(j)} + \phi(b_k^{(j)} - a_k^{(j)}),$$

and evaluates f at all 2^n points in the Cartesian product $\mathbf{x}_1 \times \mathbf{x}_2$. Let $\mathbf{m}$ be the point attaining the smallest function value. The update rule replaces, for each coordinate, the boundary opposite to the corresponding component of $\mathbf{m}$. Hence, $\Omega_{k+1}^{(n)}$ is obtained by replacing boundary components of $\Omega_k^{(n)}$ by interior points, implying $\Omega_{k+1}^{(n)} \subseteq \Omega_k^{(n)}$.

Invariance of the minimizer. Fix k and consider a coordinate j. The points $x_{1,j}$ and $x_{2,j}$ partition the interval $[a_k^{(j)}, b_k^{(j)}]$ into two subintervals. By the corner-monotonicity assumption, among all 2^n candidate points, the point $\mathbf{m}$ minimizing f is the one that is coordinatewise closest to $\mathbf{x}^*$. Consequently, for each coordinate j, the retained subinterval chosen by the algorithm contains x_j^*. Since this holds for all coordinates, we conclude that $\mathbf{x}^* \in \Omega_{k+1}^{(n)}$.

Geometric shrinkage. For each coordinate j, the retained subinterval has length

$$b_{k+1}^{(j)} - a_{k+1}^{(j)} = \phi\,(b_k^{(j)} - a_k^{(j)}),$$

where $\phi = (\sqrt{5}-1)/2 \in (0,1)$. Thus,

$$b_k^{(j)} - a_k^{(j)} = \phi^k\,(b_0^{(j)} - a_0^{(j)}) \xrightarrow[k\to\infty]{} 0.$$

Convergence. The sets $\Omega_k^{(n)}$ form a nested sequence of nonempty closed boxes with

$$\operatorname{diam}(\Omega_k^{(n)}) = \|\mathbf{b}_k^{(n)} - \mathbf{a}_k^{(n)}\| \to 0.$$

Therefore, their intersection consists of a single point. Since $\mathbf{x}^* \in \Omega_k^{(n)}$ for all k, this point must be $\mathbf{x}^*$. Hence, $\mathbf{a}_k^{(n)} \to \mathbf{x}^*$ and $\mathbf{b}_k^{(n)} \to \mathbf{x}^*$ as $k \to \infty$. □

3.2 Comparative Analysis

The proposed algorithm is compared with other gradient-based algorithms such as BFGS, CG, and AGD with the same momentum coefficient of $\theta = 0.95$. The numerical experiments of this paper were implemented in Python 3.7.6 64-bit from the Anaconda distribution using

a computer with Intel® Core™ i5-8300H CPU @ 2.30 GHz processor, 19.9 GB usable RAM, and 64-bit Microsoft Windows 10 operating system.

We used five benchmark functions named the Bohachevsky No. 1, Booth, Easom, Ackley, and Rastrigin functions. These initialized with 2500 different points uniformly sampled from their given bounds. The outcomes of 2500 minima correspond to percentages of their global convergence. The results of the optimization for each algorithm are in Figure 3.

For the Bohachevsky No. 1 function (abbreviated as Boha. No. 1), AGD-nGSS tallied 100.00% convergence compared to 17.72% of BFGS, 30.68% of CG, and 65.68% of AGD. Meanwhile, all algorithms achieved 100% while minimizing the Booth function. Yet, the mean function value of AGD-nGSS has the lowest order of magnitude at -13, beating the results of BFGS, CG, and AGD with -10, -11, and -10 orders of magnitude, respectively.

As for the remaining functions, AGD-nGSS achieved the best percentage results. It tallied 45.20% for the Easom function while the other algorithms only converged when the initial point is placed directly at the global minimum, corresponding to 0.04%. The algorithm also yielded 100.00% convergence for Ackley (BFGS: 1.28%, CG: 0.68%, and AGD: 6.16%) and 97.68% for Rastrigin (BFGS: 1.60%, CG: 1.28%, and AGD: 2.40%).

The structure and shape of the function can pertain to how AGD-nGSS performs. Ackley, Bohachevsky No. 1 and Rastrigin functions are highly multimodal with near distances between their stationary points. The radius size played a role in the minimization of the Easom function. As for the Booth function, the precision of convergence stems from the inclusion of the local search. These observations show the versatility of the algorithm depending on how to approach the problem.

In this section, we consider four functions chosen for their multimodality and stationary point placements. The Dixon-Price function has two global minima at a nearby saddle point while the Du et al. function [30] passes through two saddle points. Additionally, Beale and McCormick functions have a located starting point closer to a local minimum.

For the case of the Du et al. function, the values of the parameters used are $L = 1$, $\gamma = 1$, and $\tau = e$. Due to its construction, the method of numerical differentiation used is the five-point forward formula. For more details about this function, we refer the reader to [30].

These problems were initialized at their saddle points and observed their convergence. These are the points in Table 2. The Du et al. function is built with a chosen starting point to pass through two saddle points. The result of the trajectories from the starting point to convergence is in Figure 4.

The algorithm escaped the saddle points of the Du et al. function. On the other hand, it immediately approached one of the global minima for the Dixon-Price function. Even with nearby local minima in McCormick and Beale functions, the algorithm proceeds to the global minimum.

In general, our algorithm managed to converge for all four functions and determined the global minima of each of these functions with function values satisfying Inequality (6). Since the initial point has its gradient norm within tolerance, BFGS, CG, and AGD remained at

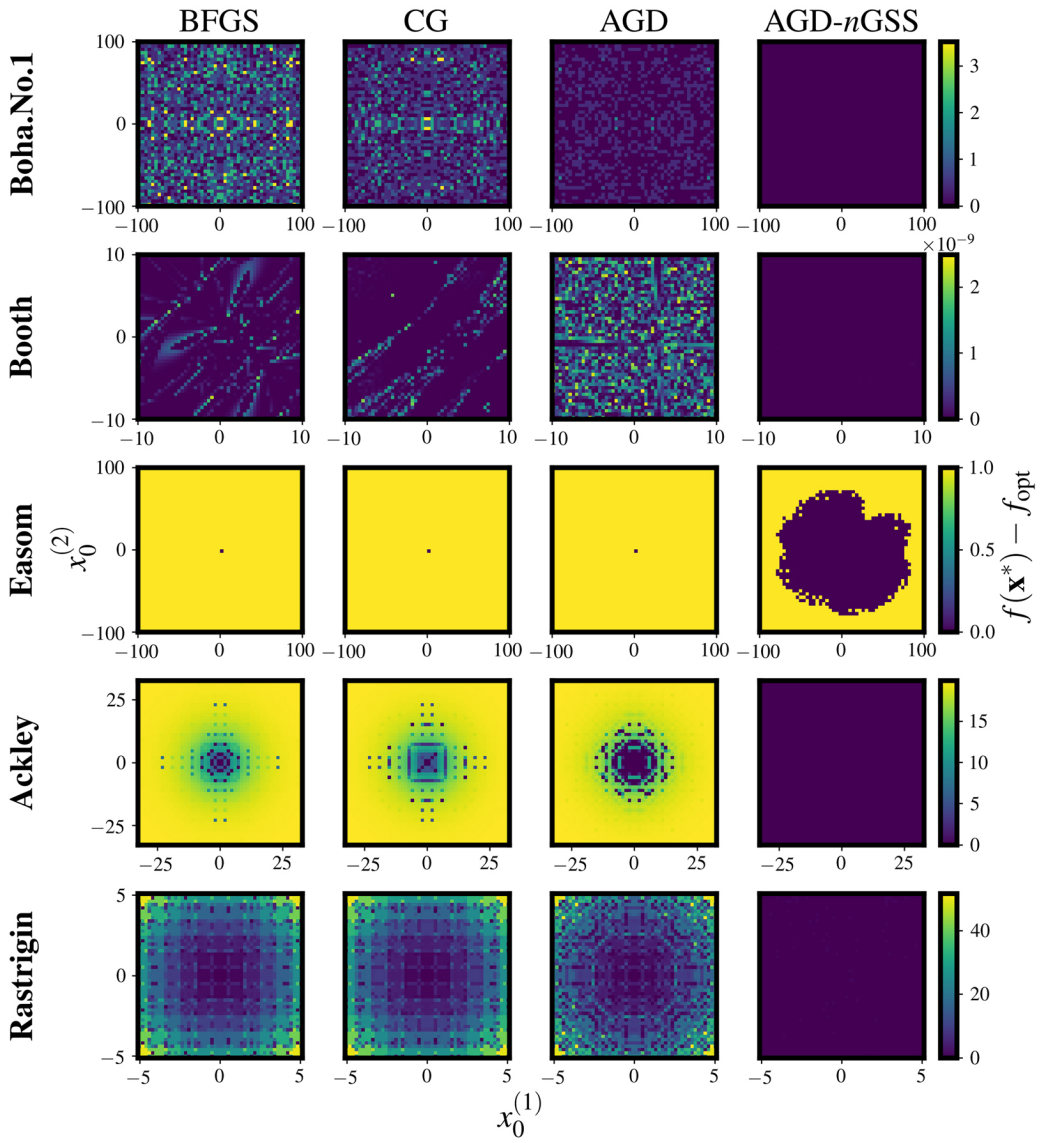

Figure 3: **Varying initial point matrices of convergence results for the benchmark functions**. For the functions Bohachevsky No. 1, Booth, Easom, Ackley, and Rastrigin, 2500 initial values were generated by combinations of 50 x-coordinates and 50 y-coordinates to obtain 2500 minimizers for each algorithm. Shown are the values of the difference between the function value of the minimizers and the global minimum. A value close to zero implies global convergence.

Table 2: Stationary points of the selected functions for the numerical experiments.

Function	Saddle Points	Global Minimizer
McCormick	$(1.54719, 0.54719)$	$(-0.54719, -1.54719)$
Dixon-Price	$\left(\frac{1}{3}, 0\right)$	$\left(1, \frac{\sqrt{2}}{2}\right)$, $\left(1, -\frac{\sqrt{2}}{2}\right)$
Beale	$(0, 1)$	$(3, 0.5)$
Du et al.	$(0, 0)$, $(4e, 0)$	$(4e, 4e)$

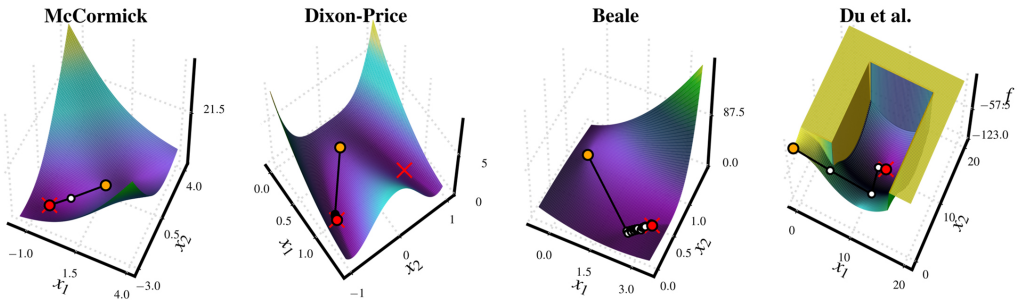

Figure 4: **Trajectories of AGD-nGSS for the functions escaping stationary points**. Shown are the 3d plots of McCormick, Dixon-Price, Beale, and Du et al. functions with iterations of AGD-nGSS using their respective initial points (orange filled). The algorithm converged globally (red filled) for all as seen approaching the global minimum (red cross/es).

the saddle point.

3.3 Application on Matrix Decomposition

Given a square matrix $A \in \mathbb{R}^{n\times n}$, low-rank matrix approximation determines a matrix $X \in \mathbb{R}^{n\times k}$, $1 \leq k \leq \text{rank}(A)$ that satisfies

$$\min_{X\in\mathbb{R}^{n\times k}} f(X) = \min_{X\in\mathbb{R}^{n\times k}} \left|\left|XX^T - A\right|\right|_F, \tag{7}$$

where $||\cdot||_F$ is the Frobenius norm. Equation (7) is highly nonconvex [31], and gradient-based methods have been subjected to answer these types of problems [32].

An exact gradient vector

$$\nabla f(X) = 4XX^TX - 2(A^T + A)X,$$

for the problem in Equation (7) has been proven in [33]. It is obvious from this expression that the zero matrix is a stationary point which is also a saddle point of the optimization problem in Equation (7) [5]. We use this as the initial guess of the algorithms for the experiments. For example, consider

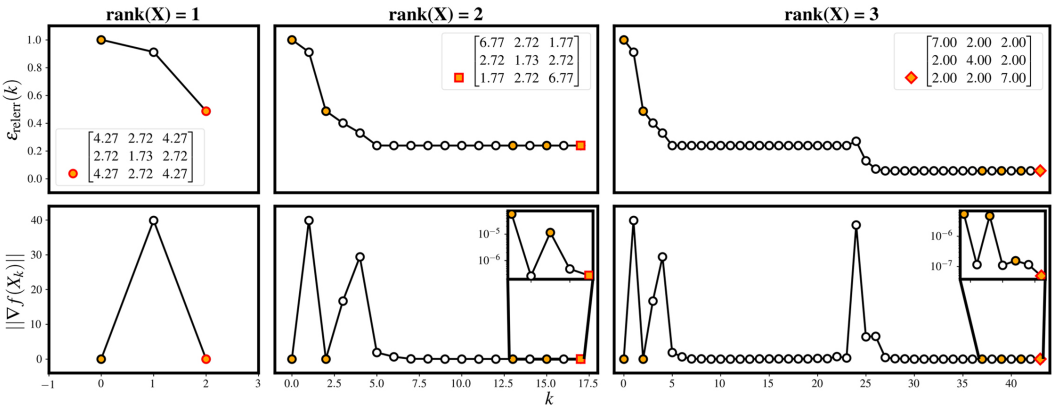

Figure 5: **Iterations of matrix decomposition for each rank of AGD-nGSS**. For AGD-nGSS, starting at the zero matrix, the (first row) relative residual error and (second row) gradient norm are determined for each iteration. The local search was triggered at suspected stationary points (orange dot) and converged (red edges) with the final approximation as the label.

$$A = \begin{bmatrix} 7 & 2.1 & 2.3 \\ 1.9 & 4 & 1.65 \\ 1.7 & 2.35 & 7. \end{bmatrix} \tag{8}$$

The measures to be observed are the norm of the gradient of the minimizer and the relative residual error. The latter is given by Equation (9),

$$\varepsilon_{\text{relerr}}(k) = \frac{||X_k X_k^T - A||_F}{||A||_F}, \tag{9}$$

where k is the iteration of the minimization. As given by the expression, the metric is a normalized function value that compares the relative error between the approximations and the true matrix.

For the problem in Equation (8), we used three rank values to obtain approximations of matrix decomposition. By initializing at the zero matrix, BFGS and CG did not obtain minimizers for all configurations. As for AGD-nGSS, the optimization results are shown in Figure 5.

The rank-1 minimization is a three-dimensional problem while rank-2 is six-dimensional, and rank-3 is nine-dimensional. As the problem increases the rank of the minimizer, the number of iterations tends to increase as well. With these increasing ranks of the optimization problem, AGD-nGSS was still able to give convergent results. The rank-1 converged only after three steps while it took 18 and 44 steps for rank-2 and rank-3, respectively.

The modified local search activated five times for the case of rank-2 and six times for rank-3. The last few of these triggers appeared to be almost minimum, but the algorithm proceeded further to satisfy the tolerance. This occurrence highlights the importance of the

local search as a greedy validator of the algorithm.

Even with a gradient tolerance of 10^{-4}, the obtained minimizers were less than this value in their respective orders of magnitude that can be pertained to the lesser numerical tolerance as the stopping criterion of the modified local search. Consequently, the relative residual error decreased in value as the rank of the problem increased. AGD-nGSS not only manages to escape the zero matrix but also converges to proper matrices for factorization.

3.4 Parameter Estimation of a Biological Model

Parameter estimation determines the best set of parameter values for a model corresponding to a given data. For this paper, we will use our algorithm to identify parameters of the Fitzhugh-Nagumo model from generated data.

The Fitzhugh-Nagumo model is a mathematical model based on the individual neuronal researches of Fitzhugh [34] and Nagumo et al. [35]. It is a system of first-order ordinary differential equations dependent on two variables, the membrane potential v and the recovery variable r, given by

$$\frac{dv}{dt} = c\left(v - \frac{v^3}{3} + r\right) \text{ and } \frac{dr}{dt} = -\frac{v - a + br}{c}. \tag{10}$$

The equations consist of two variables v and r and three nonlinear parameters a, b, and c. These solutions exhibit reciprocity due to their coupled relationship as the voltage and the feedback current.

The chosen approach of parameter estimation is the mean squared difference method. The method attempts to adjust the parameters that make the error between the data and the model small enough. This loss function for the model is described by the following expression,

$$\min_{\phi} \frac{1}{n} \sum_{i=1}^{n} \left[\left(\hat{V}_i - V_i(\phi)\right)^2 + \left(\hat{R}_i - R_i(\phi)\right)^2 \right], \tag{11}$$

where $\hat{V}_i$ is the ith data point for the variable v, $\hat{R}_i$ is the ith data point for the variable r, $V_i(\phi)$ is the solution of v corresponding to iteration i, $R_i(\phi)$ is the solution of r corresponding to iteration i, ϕ is the parameter set applied to the model Y, and n is the total number of data points. The composition of this specific problem is based on [36, 37]. The chosen model consists of three parameters a, b, and c, but in this study, the loss function only deals with the two parameters a and b.

Sets of noisy solutions resemble data using the true parameters $\{a^*, b^*, c^*\} = \{0.2, 0.2, 3.0\}$ to solve the model. We simulated solutions with random noise drawn from a Gaussian distribution with varying values of the standard deviation to represent the spread of the noise. The noise added to the solutions is leveled using standard deviation values from 0.125 (or 12.5%) up to 1.00 (100.0%).

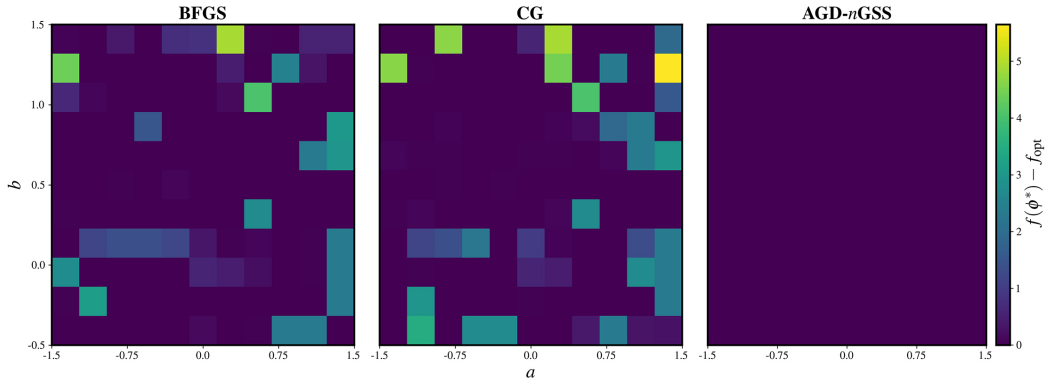

Figure 6: **Varying initial point matrices of convergence results for the parameter estimation of the Fitzhugh-Nagumo model**. Each algorithm determined their respective minimum for the parameter estimation of the Fitzhugh-Nagumo model on 121 initial values. The noise level considered to generate the data is held at $\sigma = 0.5$. Shown are the values of the difference between the function value of the minimizers and the global minimum. A value close to zero implies global convergence.

Table 3: Determined function values for varying the initial point of the parameter estimation of the Fitzhugh-Nagumo model.

Algorithm	%	**Function value**		
		Mean	Std	Median
BFGS	43.80	0.98	1.01	0.51
CG	39.67	1.17	1.28	0.52
AGD-nGSS	100.00	0.50	1.76E-07	0.50

For the solutions, the time frame considered is 2000 ms, and the time step is 0.05 ms. This configuration corresponds to 400 data points for each solution and each data set. The initial value of V and R is held at -1.0 and 1.0, respectively.

The desired global minimum for each algorithmic run is at the set of true parameters $\phi^* = \{a^*, b^*\} = \{0.2, 0.2\}$. Therefore, the gradient tolerance is readjusted as $\eta_{\text{grad}} = ||\nabla f(\phi^*)||$ and the numerical tolerance is held at $\eta_{\text{num}} = 10^{-3}$. We will compare results for BFGS, CG, and the proposed algorithm, AGD-nGSS.

First, we vary the starting points of the algorithm for this problem. Initial point matrices can then represent the problem as an 11×11 matrix for a total of 121 algorithm runs. The chosen noise level is at $\sigma = 0.5$ for the data generation. The results are illustrated in Figure 6 and tabulated in Table 3.

Figure 6 gives a clear and decisive result in favor of AGD-nGSS. With a specified tolerance of 10^{-2}, the algorithm performed well, with a 100% of the runs converging successfully. The BFGS and CG algorithms achieved 43.80% and 39.67% percentage of successful runs, respectively.

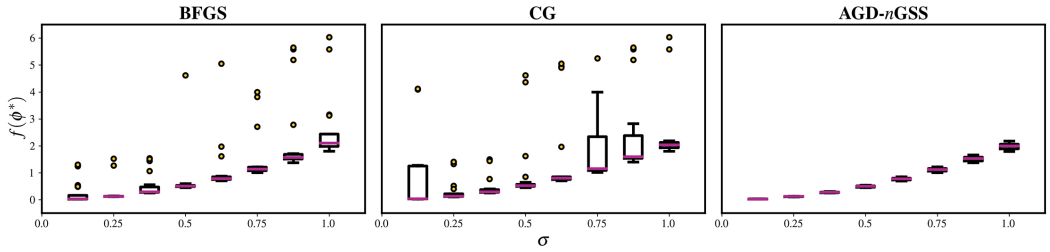

Figure 7: **Resulting box plots of the function values for varying noise levels**. Trends of the function value in terms of the median, interquartile ranges, and minimum and maximum values are shown for (left) BFGS, (center) CG, and (right) AGD-nGSS as the generated noise level increases.

The results in Table 3 show that AGD-nGSS assured convergence to a minimizer. The induced standard deviation was at a value close to zero. These observations on the statistical metrics signify that even with changing initialization of the method, AGD-nGSS can still determine the precise set of parameters for the model.

The applied noise to generate the data was also varied from $\sigma = 0.125$ to $\sigma = 1.000$. For each of the σ values, twenty data sets were reproduced and used to identify parameters.

Figure 7 represents the result of the optimization in terms of the median and interquartile ranges of the function values via box plots. For all algorithms, the function values escalate as the noise level increase. Interestingly, the spread of obtained quantities of AGD-nGSS showed much tighter interquartile ranges with no outliers. Variation of the noise level did not hinder the algorithm from properly converging towards the assigned parameters. This result shows the resilience of AGD-nGSS amidst such conditions.

4 Conclusions

The proposed algorithm is called the *accelerated gradient descent with the n-dimensional golden section search* or AGD-nGSS. The two-stage method uses Nesterov's AGD as the global search that travels the problem landscape and the gradient-free modified nGSS to verify convergence.

We tested the algorithm on benchmark functions, particularly problems containing nonoptimal stationary points, to show its effectiveness in escaping these points towards the global minimum. The selected functions are initialized on 2500 starting points to show the percentage of convergence of the algorithm. AGD-nGSS generated a high percentage of successful runs for 2500 starting points with highly minimal function values.

To position the proposed method relative to existing optimization strategies, Table 4 summarizes key advantages and limitations of AGD–nGSS in comparison with representative classes of algorithms. In contrast to classical gradient-based methods, AGD–nGSS incorporates a principled local mechanism to validate stationary points. Compared to meta-

heuristic approaches, the proposed method retains deterministic behavior and faster convergence. These characteristics make AGD–nGSS a balanced alternative for problems where gradient information is available but local nonconvexity poses challenges.

Table 4: Qualitative comparison of AGD–nGSS with existing optimization methods.

Method	**Uses Gradients**	**Escapes Local Traps**	**Deterministic**
Gradient Descent / AGD	Yes	Limited	Yes
Metaheuristic Methods	No	Yes	No
Second-Order Methods	Yes (Hessian)	Yes	Yes
AGD–nGSS (proposed)	Yes (global), No (local)	Yes	Yes

We also applied the method to the low-rank matrix approximation, which determines a matrix that can successfully factorize a given matrix. For all ranks of the minimizer, only the proposed algorithm ended up with appropriate matrices with gradient norm values approaching zero and a minimal relative residual error.

Parameter estimation is done for a simplistic neuronal system known as the Fitzhugh-Nagumo model. By varying the initial guess, AGD-nGSS accomplished global convergence for all 121 points, performing better than BFGS and CG. The noise level of the data was analyzed for all independent runs, resulting in minimal function values with high precision due to their narrow standard deviations. We observed that AGD-nGSS is capable of precisely determining minimizers amidst varying conditions.

With these results, AGD-nGSS achieves effective performance that withstands the discussed limitations of gradient-based methods. The proposed approach can converge globally despite being deterministic. There are many possibilities to improve the algorithm via generalized line search methods, variability of parameters, and modifications of the local search. A modification of the proposed scheme to handle high-dimensional problems is also an exciting research direction.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] C. Lemaréchal, *Cauchy and the gradient method*, Documenta Mathematica 251 (2012) 254.

[2] J. McKeown, D. Meegan, D. Sprevak, *An introduction to unconstrained optimisation*, CRC Press, 1990.

[3] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate*

of convergence o (1/k^ 2), in: Proceedings of the USSR Academy of Sciences, volume 269, 1983, pp. 543–547.

[4] Y. Nesterov, *Lectures on convex optimization*, volume 137, Springer, 2018.

[5] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, M. I. Jordan, *How to escape saddle points efficiently*, in: International Conference on Machine Learning, PMLR, 2017, pp. 1724–1732.

[6] C. Jin, P. Netrapalli, M. I. Jordan, *Accelerated gradient descent escapes saddle points faster than gradient descent*, in: Conference on Learning Theory, 2018, pp. 1042–1085.

[7] S. Reddi, M. Zaheer, S. Sra, B. Poczos, F. Bach, R. Salakhutdinov, A. Smola, *A generic approach for escaping saddle points*, in: International Conference on Artificial Intelligence and Statistics, 2018, pp. 1233–1242.

[8] A. Anandkumar, R. Ge, *Efficient approaches for escaping higher order saddle points in non-convex optimization*, in: Conference on Learning Theory, 2016, pp. 81–102.

[9] C. Fang, Z. Lin, T. Zhang, *Sharp analysis for nonconvex sgd escaping from saddle points*, in: Conference on Learning Theory, PMLR, 2019, pp. 1192–1234.

[10] M. M. Noel, *A new gradient based particle swarm optimization algorithm for accurate computation of global minimum*, Applied Soft Computing 12 (2012) 353–359.

[11] P. K. Gudla, R. Ganguli, *An automated hybrid genetic-conjugate gradient algorithm for multimodal optimization problems*, Applied Mathematics and Computation 167 (2005) 1457–1474.

[12] W. Xie, W. Yu, X. Zou, *Diversity-maintained differential evolution embedded with gradient-based local search*, Soft Computing 17 (2013) 1511–1535.

[13] W. Jiang, S. Siddiqui, *Hyper-parameter optimization for support vector machines using stochastic gradient descent and dual coordinate descent*, EURO Journal on Computational Optimization 8 (2020) 85–101.

[14] J. A. Koupaei, S. M. M. Hosseini, F. M. Ghaini, *A new optimization algorithm based on chaotic maps and golden section search method*, Engineering Applications of Artificial Intelligence 50 (2016) 201–214.

[15] G. Cocchi, M. Lapucci, P. Mansueto, *Pareto front approximation through a multi-objective augmented lagrangian method*, EURO Journal on Computational Optimization (2021) 100008.

[16] J. Kiefer, *Sequential minimax search for a maximum*, Proceedings of the American Mathematical Society 4 (1953) 502–506.

[17] C. He, Y. F. Zheng, S. C. Ahalt, *Object tracking using the gabor wavelet transform and the golden section algorithm*, IEEE Transactions on Multimedia 4 (2002) 528–538.

[18] M. A. Rahman, S. Liu, S. Lin, C. Wong, G. Jiang, N. Kwok, *Image contrast enhancement for brightness preservation based on dynamic stretching*, International Journal of Image Processing 9 (2015) 241.

[19] D. H. Yeom, J. B. Park, Y. H. Joo, *Selection of coefficient for equalizer in optical disc drive by golden section search*, IEEE Transactions on Consumer Electronics 56 (2010) 657–662.

[20] Y. C. Chang, *N-dimension golden section search: Its variants and limitations*, in: 2009 2nd International Conference on Biomedical Engineering and Informatics, IEEE, 2009, pp. 1–6.

[21] P. K. Salonga, J. M. Inaudito, R. Mendoza, *An unconstrained minimization technique using successive implementations of golden search algorithm*, in: AIP Conference Proceedings, volume 2192, 2019, p. 060018.

[22] B. Addis, M. Locatelli, *A new class of test functions for global optimization*, Journal of Global Optimization 38 (2007) 479–501.

[23] S. Surjanovic, D. Bingham, *Virtual library of simulation experiments: Test functions and datasets*, Retrieved December 16, 2020, from `http://www.sfu.ca/~ssurjano`, 2013.

[24] T. Aonishi, R. Maruyama, T. Ito, H. Miyakawa, M. Murayama, K. Ota, *Imaging data analysis using non-negative matrix factorization*, Neuroscience Research 179 (2022) 51–56.

[25] L. Ou-Yang, F. Lu, Z.-C. Zhang, M. Wu, *Matrix factorization for biomedical link prediction and scrna-seq data imputation: an empirical survey*, Briefings in Bioinformatics 23 (2022) bbab479.

[26] H. Liu, C. Zheng, D. Li, X. Shen, K. Lin, J. Wang, Z. Zhang, Z. Zhang, N. N. Xiong, *Edmf: Efficient deep matrix factorization with review feature learning for industrial recommender system*, IEEE Transactions on Industrial Informatics 18 (2021) 4361–4371.

[27] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, J. Cheng, *A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models*, Energy Conversion and Management 230 (2021) 113784.

[28] X. Chen, J. Li, C. Xiao, P. Yang, *Numerical solution and parameter estimation for uncertain sir model with application to covid-19*, Fuzzy Optimization and Decision Making 20 (2021) 189–208.

[29] C. U. Jamilla, R. G. Mendoza, V. M. P. Mendoza, *Parameter estimation in neutral delay differential equations using genetic algorithm with multi-parent crossover*, IEEE Access 9 (2021) 131348–131364.

[30] S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, A. Singh, B. Poczos, *Gradient descent can take exponential time to escape saddle points*, in: Advances in Neural Information Processing Systems, 2017, pp. 1067–1077.

[31] Y. Chi, Y. M. Lu, Y. Chen, *Nonconvex optimization meets low-rank matrix factorization: An overview*, IEEE Transactions on Signal Processing 67 (2019) 5239–5269.

[32] R. A. Pitaval, W. Dai, O. Tirkkonen, *Convergence of gradient descent for low-rank matrix approximation*, IEEE Transactions on Information Theory 61 (2015) 4451–4457.

[33] X. Duan, J. Li, Q. Wang, X. Zhang, *Low rank approximation of the symmetric positive semidefinite matrix*, Journal of Computational and Applied Mathematics 260 (2014) 236–243.

[34] R. FitzHugh, *Impulses and physiological states in models of nerve membrane*, Biophysical Journal 1 (1961) 445–466.

[35] J. Nagumo, S. Arimoto, S. Yoshizawa, *An active pulse transmission line simulating nerve axon*, Proceedings of the Institute of Radio Engineers 50 (1962) 2061–2070.

[36] J. O. Ramsay, G. Hooker, D. Campbell, J. Cao, *Parameter estimation for differential equations: a generalized smoothing approach*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69 (2007) 741–796.

[37] J. Calver, *Parameter estimation for systems of ordinary differential equations*, Ph.D. thesis, University of Toronto, 2019.

This page is intentionally left blank