mathematics

Article

Comparative Analysis of Accelerated Models for Solving Unconstrained Optimization Problems with Application of Khan's Hybrid Rule

Vladimir Rakočević [1,2] and Milena J. Petrović [3,*]

[1] Serbian Academy of Sciences and Arts, Kneza Mihajla 35, 11000 Belgrade, Serbia
[2] Faculty of Sciences and Mathematics, University of Niš, Višegradska 33, 18106 Niš, Serbia
[3] Faculty of Sciences and Mathematics, University of Pristina in Kosovska Mitrovica, Lole Ribara 29, 38220 Kosovska Mitrovica, Serbia
[*] Correspondence: milena.petrovic@pr.ac.rs; Tel.: +381-28-425-396

Abstract: In this paper, we follow a chronological development of gradient descent methods and its accelerated variants later on. We specifically emphasise some contemporary approaches within this research field. Accordingly, a constructive overview over the class of hybrid accelerated models derived from the three-term hybridization process proposed by Khan is presented. Extensive numerical test results illustrate the performance profiles of hybrid and non-hybrid versions of chosen accelerated gradient models regarding the number of iterations, CPU time, and number of function evaluation metrics. Favorable outcomes justify this hybrid approach as an accepted method in developing new efficient optimization schemes.

Keywords: line search; gradient descent methods; quasi-Newton method; convergence rate

MSC: 49M15; 49M37; 65B99; 90C26; 90C30; 90C53

check for updates

Citation: Rakočević, V.; Petrović, M.J. Comparative Analysis of Accelerated Models for Solving Unconstrained Optimization Problems with Application of Khan's Hybrid Rule. *Mathematics* **2022**, *10*, 4411. https://doi.org/10.3390/math10234411

Academic Editor: Simeon Reich

Received: 30 October 2022
Accepted: 21 November 2022
Published: 23 November 2022

1. Class of Accelerated Gradient Descent Methods and Its Benefits

Many contemporary scientific, engineering, medical and problems from various other research areas, are closely related to mathematical optimization theory. Among all others, the unconstrained optimization problems are the most frequently considered [1–8]. Owing to the duality principle, an optimization problem may be viewed as a minimization problem. Unconstrained minimization problems can simply be stated as finding

$$\min f(x),\ x \in \mathbb{R}^n,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function, that is solved by the general iteration:

$$x_{k+1} = x_k + t_k d_k. \tag{1}$$

In (1), x_k presents the current iterative point, x_{k+1} is the next one; the positive iterative step length value is denoted by t_k while d_k stands for the k-th search direction vector. As can be observed, two main elements that measure the efficiency and robustness of the iterative rule (1) are the adequately calculated iterative step size t_k and the properly chosen iterative search direction d_k. Since we are dealing with minimization problems, it is a natural choice to define the search direction so that it fulfils the descent condition, i.e.,

$$g_k^T d_k < 0, \tag{2}$$

where g_k is the gradient of f at the point x_k. Apart from that, we use standard notations for the gradient and the Hessian of the objective function f:

$$g(x) = \bigtriangledown f(x), \quad G(x) = \bigtriangledown^2 f(x), \quad g_k = \bigtriangledown f(x_k), \quad G_k = \bigtriangledown^2 f(x_k). \tag{3}$$

Following condition (2), it is easy to conclude that the $d_{\equiv} - g_k$ produces the most certain descent direction, known as the gradient descent direction. Iteration (1) that includes the gradient descent direction is known as the gradient descent method (or GD method)

$$x_{k+1} = x_k - t_k g_k. \tag{4}$$

Step length parameter t_k in iterations (1) and (4), is determined either by the exact or by some of the inexact line search procedures. Using the exact line search technique, iterative step length value t_k is computed by solving the following minimization task:

$$f(x_k + t_k d_k) = \min f(x_k + td_k), \quad t > 0. \tag{5}$$

It is clear that solving the previous minimization problem in each iterative value presents a time- and resource-consuming task regarding CPU time requirements and the number of required objective function evaluations. For this reason, most contemporary optimization methods use the inexact line search algorithms to calculate iterative step size instead of the exact line search procedure. The convergence properties of line search methods for unconstrained optimization are specifically examined in [9]. Some of the frequently used inexact line search algorithms are weak and strong Wolfe's algorithms [10], Backtracking algorithm proposed in [11] with Armijo's rule [12], etc.:

Weak Wolfe's line search:

$$f(x_k + t_k d_k) \leq f(x_k) + \delta t_k g_k^T d_k$$
$$g(x_k + t_k d_k)^T d_k \geq \sigma g_k^T t d_k;$$

Strong Wolfe's line search:

$$f(x_k + t_k d_k) \leq f(x_k) + \delta t_k g_k^T d_k$$
$$|g(x_k + t_k d_k)^T d_k| \geq -\sigma g_k^T t d_k;$$

Backtracking algorithm:

1. Objective function $f(x)$, the direction d_k of the search at the point x_k and numbers $0 < \sigma < 0.5$ and $\beta \in (0, 1)$ are required;
2. $t = 1$;
3. $f(x_k + td_k) > f(x_k) + \sigma t g_k^T d_k$, take $t := t\beta$;
4. Return $t_k = t$.

Subsequently, the Newton method with line search is given as

$$x_{k+1} = x_k - t_k G_k^{-1} g_k. \tag{6}$$

In (6), G_k^{-1} stands for the inverse of the function Hessian, according to the previously adopted notation. Step length parameter t_k is obtained by applying some chosen inexact procedure. Instead of calculating the inverse of the function Hessian, which is often a demanding task, in quasi-Newton methods the adequate approximation of the Hessian (or of its inverse) is used

$$x_{k+1} = x_k - t_k H_k g_k. \tag{7}$$

Herein, $H_k \equiv B_k^{-1}$ and B_k is derived as a symmetric positive definite Hessian's approximation. Updating of $\{B_i\}$, $i \in \mathbb{N}$ is conducted using the quasi-Newton property of secant equation

$$B_{k+1} s_k = y_k,$$

where parameters s_k and y_k are the differences in two successive iterative points and iterative gradients, respectively, i.e., $s_k = x_{k+1} - x_k, \quad y_k = g_{k+1} - g_k$. In [13], the classification of methods for updating the matrix B_k is presented. Therein, the three updating methods are extricated:

1. matrix B_k is defined as a scalar matrix, i.e., $B_k = \gamma_k I, \gamma_k > 0$;
2. matrix B_k is defined as a diagonal matrix, i.e., $B_k = diag(\lambda_1, \cdots, \lambda_n), \lambda_i > 0, i = \overline{1, n}$;
3. matrix B_k is defined as a full matrix.

Taking the simplest updating approach, the first one, i.e., $B_k = \gamma_k I \backsim G_k, \gamma_k > 0$, the quasi-Newton method (7) is transformed to

$$x_{k+1} = x_k - t_k \gamma_k^{-1} g_k. \tag{8}$$

In [14], the authors named the iterative methods (8) as a *class of accelerated gradient methods*. They named it so for their good convergence and performance metrics. Previously in [15], Andrei defined this accelerated iteration by calculating parameter $\theta_k (= \gamma_k^{-1})$ as follows:

Algorithm for generating scalar θ_k from [15]:

1. Objective function $f(x)$, the direction d_k of the search at the point x_k and numbers $0 < \sigma < 0.5$ and $\beta \in (0,1)$ are required;
2. Apply Backtracking algorithm to calculate $t_k \in (0, 1]$;
3. Compute $z = x_k - t_k g_k, \quad g_z = \nabla f(z), \quad y_k = g_z - g_k$;
4. Compute $a_k = t_k g_k^T g_k, \quad b_k = -t_k y_k^T g_k$
5. Return $\theta_k = \frac{a_k}{b_k}$.

Stanimirović and Miladinović in [14] determined the parameter γ_k^{-1} from (8) on the basis of the second-order Taylor's expansion, and denoted it as SM method. Results obtained by several researchers on this topic confirmed that this way of deriving the accelerated parameter (as named in [16]), is justifiable regarding the convergence and numerical performance aspects [17,18]. Several forms of this important variable in chosen accelerated gradient schemes are listed as expressions (5)–(9) in [19] and some other approaches are presented in [20–22].

Among the various accelerated gradient iterations, for this research we specifically chose three among which Khan's hybridization three-term process was later applied. The first of the three is the already mentioned SM method, defined by relation (8) and presented in [14]. The second one is the ADD method (i.e., *accelerated double direction method*) from [16], with the iterative representation

$$x_{k+1} = x_k + \alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k. \tag{9}$$

In (9), $\alpha_k > 0$ is the iterative step length value, while d_k is the second search vector calculated under the assumption $\|d_k\| = 1, k = 1, 2, \ldots$, by the next procedure:

$$d_k(t) = \begin{cases} d_k^*, & k \leq m - 1 \\ \sum_{i=2}^{m} t^{i-1} d_{k-i+1}^* & k \geq m \end{cases} \tag{10}$$

and d_k^* is the solution of the problem $\min_{x \in \mathbb{R}} \Phi_k(d)$,

$$\Phi_k(d) = \bigtriangledown f(x_k)^T d + \frac{1}{2}\gamma_{k+1} I = g(x_k)^T d + \frac{1}{2}\gamma_{k+1} I. \tag{11}$$

The iterative value of the accelerated parameter γ_k, obtained through Taylor's series of (9), is

$$\gamma_{k+1}^{ADD} = 2\frac{f(x_{k+1}) - f(x_k) - \alpha_k g_k^T \left(\alpha_k d_k - \gamma_k^{-1} g_k\right)}{\left(\alpha_k d_k - \gamma_k^{-1} g_k\right)^T \left(t_k d_k - \gamma_k^{-1} g_k\right)}. \tag{12}$$

The positive step length value α_k of the ADD scheme is derived using the Backtracking algorithm, starting with initial $t = 1$. Taking the following substitutions in (9)

$$\alpha_k^2 \to \beta_k,$$

where β_k is calculated by a different Backtracking procedure, and

$$d_k \to -g_k,$$

lead to the $ADSS$ (accelerated double step size) method [17]:

$$x_{k+1} = x_k - \alpha_k \gamma_k{}^{-1} g_k - \beta_k g_k = x_k - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) g_k. \tag{13}$$

Finally, under assumption

$$\alpha_k + \beta_k = 1,$$

$ADSS$ iteration is transformed to $TADSS$ scheme [18], our third chosen accelerated gradient descent model:

$$x_{k+1} = x_k - [\alpha_k(\gamma_k{}^{-1} - 1) + 1] g_k. \tag{14}$$

As in (9), the iterative step length value α_k in (14) is calculated on the basis of Backtracking algorithm. The accelerated parameter of the $TADSS$ scheme is

$$\gamma_{k+1}^{TADSS} = 2\frac{f(x_{k+1}) - f(x_k) + \psi_k \|g_k\|^2}{\psi_k^2 \|g_k\|^2}, \quad \psi_k = [\alpha_k \gamma_k^{-1} - \alpha_k^2) + 1]. \tag{15}$$

In the following proposition, we prove that (8), (9) and (14) iterations are a gradient descending process.

Proposition 1. *The search directions in iterations defined by* (8), (9) *and* (14) *fulfil the descending condition* (2).

Proof. We separately analyze the search directions of all three listed methods.

- According to the general iteration form (1), the search direction in SM method, defined by relation (8), is $d_k \equiv -\gamma_k^{-1} g_k$. One of the essential properties of the accelerated parameter γ_k is its positiveness. If in some iterative step k of the accelerated gradient algorithms with leading iterative rules (8), (9) and (14) this necessary condition is not fulfilled, then the k-th accelerated scalar value is set to be $\gamma_k = 1$. Bearing this fact in mind, we easily conclude that

$$\mathbf{g}_k^T \mathbf{d}_k = \mathbf{g}_k^T(-\gamma_k^{-1}\mathbf{g}_k) = -\gamma_k^{-1}\|\mathbf{g}_k\|^2 < 0,$$

  which confirms that the condition (2) in SM method, defined by (8), is fulfilled.
- Accelerated double direction ADD scheme (9) contains two search vectors. The first one, denoted as d_k, is defined by (10). The second one is of the same form as in the SM iteration, i.e., $-\gamma_k{}^{-1} g_k$. In the procedure (10), crucial element d_k^* in deriving vector d_k is defined as a solution of the minimization problem (11) that depends on the gradient g_k, under the assumption $\|d_k\| = 1$. Thus, the defined vector direction is a relaxed differentiable variant of the procedure for determination of the search vector d_k (rule 2) in [23] and accordingly, the $d = 0$ is globally optimum of the problem (11). Subsequently, we consider only the second direction, which is already performed in the previous item.
- In $TADSS$ scheme, vector direction can be seen as $-[\alpha_k(\gamma_k{}^{-1} - 1) + 1] g_k$. Checking the descent condition (2), we get

$$\mathbf{g}_k^T \cdot \left(-[\alpha_k(\gamma_k{}^{-1} - 1) + 1]\mathbf{g}_k\right) = -[\alpha_k(\gamma_k{}^{-1} - 1) + 1]\|\mathbf{g}_k\|^2 < 1,$$

since $\gamma_k^{-1} > 1$ and $\alpha_k \in (0,1]$ according to $TADSS$ algorithm.
□

Further on, in Section 2, we analyse Khan's hybridization rule applied on various gradient methods. Finally, Dolan–Moré representations and parallels among the hybrid and non-hybrid versions of SM, ADD and $TADSS$ schemes, obtained over large-scale numerical outcomes, are illustrated in Section 3.

2. Three-Term Khan's Hybridization Principle over the Accelerated Gradient Descent Models

For a nonempty convex subset $\mathbb{C}$ of a normed space $\mathbb{E}$, let $T : \mathbb{C} \to \mathbb{C}$ be a mapping defined on $\mathbb{C}$. Then, for some sequences u_k, v_k, z_k and y_k defined on $\mathbb{C}$, the Picard, Mann and Ishikawa iterative processes [24–26] are, respectively, given as:

$$\begin{cases} u_1 = u \in \mathbb{C}, \\ u_{k+1} = Tu_k, \quad k \in \mathbb{N}, \end{cases}$$

$$\begin{cases} v_1 = v \in \mathbb{C}, \\ v_{k+1} = (1-\alpha_k)v_k + \alpha_k T v_k, \quad k \in \mathbb{N}, \end{cases}$$

$$\begin{cases} z_1 = z \in \mathbb{C}, \\ z_{k+1} = (1-\alpha_k)z_k + \alpha_k Tyk, \\ y_k = (1-\beta_k)z_k + \beta_k T z_k. \qquad k \in \mathbb{N} \end{cases}$$

In the listed relations, parameters $\{\alpha_k\}, \{\beta_k\} \in (0,1)$ are the sequences of positive numbers, which in the Ishikawa process [26] fulfil the following assumptions

- $0 \le \alpha_k \le \beta_k \le 1, \quad k \ge 0,$
- $\lim_{k\to\infty} \beta_k = 0,$
- $\sum_{k=1}^{\infty} \alpha_k \beta_k = \infty.$

In [27], Khan proposed a new three-term iterative process as follows

$$\begin{cases} x_1 = x \in \mathbb{R}, \\ x_{k+1} = Ty_k, \\ y_k = (1-\alpha_k)x_k + \alpha_k T x_k, \quad k \in \mathbb{N} \end{cases} \tag{16}$$

with the sequence of positive numbers $\{\alpha_k\} \in (0,1)$, which is considered as a set of constant values, i.e., $(\alpha = \alpha_k \in (0,1) \quad \forall k \in \mathbb{N},)$ in practical numerical tests, as proposed in [27]. Khan confirmed in [27] that the process (16) converges faster than the processes of Picard, Mann and Ishikawa.

Khan developed this iterative process (16) as a hybrid variant of previously mentioned, well-known iterations and with that managed to improve these types of methods. Further, some authors used auspicious aspects of this hybrid rule and applied it to some accelerated gradient optimization methods. The hybridization principle consists of taking the objective accelerated iteration as a guiding operator in (16). As a result, several accelerated-hybridization processes arose [28–32]. We list them below, together with their accelerated parameters.

1. Hybrid accelerated gradient descent method (HSM) [28]

$$x_{k+1} = x_k - (\alpha_k + 1)t_k \gamma_k^{-1} g_k, \tag{17}$$

$$\gamma_{k+1}^{HSM} = 2\gamma_k \frac{\gamma_k[f(x_{k+1}) - f(x_k)] + (\alpha_k+1)t_k\|g_k\|^2}{(\alpha_k+1)^2 t_k^2 \|g_k\|^2}. \tag{18}$$

2. Hybrid accelerated double direction method ($HADD$) [29]

$$x_{k+1} = x_k - \alpha t_k \gamma_k^{-1} g_k + \alpha t_k^2 d_k, \quad \alpha \in (1,2), \tag{19}$$

$$\gamma_{k+1} = 2\frac{f(x_{k+1}) - f(x_k) - \alpha g_k^T\left(t_k^2 d_k - t_k\gamma_k^{-1}g_k\right)}{\alpha^2 t_k^2\left(t_k d_k - \gamma_k^{-1}g_k\right)^T\left(t_k d_k - \gamma_k^{-1}g_k\right)}. \tag{20}$$

3. Hybrid accelerated double step size method (*HADSS*) [30]

$$x_{k+1} = x_k - g_k\alpha(t_k\gamma_k^{-1} + p_k), \quad \alpha \equiv \alpha_k + 1 \in (1,2) \quad \forall k, \tag{21}$$

$$\gamma_{k+1}^{HADSS} = 2\frac{f(x_{k+1}) - f(x_k) + \alpha\left(t_k\gamma_k^{-1} + p_k\right)\|g_k\|^2}{\alpha^2\left(t_k\gamma_k^{-1} + p_k\right)^2\|g_k\|^2}. \tag{22}$$

4. Hybrid transformed double step size method (*HTADSS*) [31]

$$x_{k+1} = x_k - \alpha(t_k(\gamma_k^{-1} - 1) + 1)g_k, \quad \alpha \in (1,2), \tag{23}$$

$$\gamma_{k+1}^{HTADSS} = 2\frac{f(x_{k+1}) - f(x_k) + \alpha\varphi_k\|g_k\|^2}{\alpha^2\varphi_k^2\|g_k\|^2}, \tag{24}$$

where

$$\varphi_k = t_k(\gamma_k^{-1} - 1) + 1. \tag{25}$$

5. Hybrid gradient descent method (*HGD*) [32]

$$x_{k+1} = x_k - (\alpha_k + 1)t_k g_k, \quad \alpha_k \in (0,1) \quad \forall k. \tag{26}$$

6. Hybrid accelerated gradient descent method (*HAGD*) [32]

$$x_{k+1} = x_k - (\alpha_k + 1)\theta_k t_k g_k, \quad \alpha \in (0,1) \quad \forall k,\ \theta_k = \frac{\gamma_k}{t_k\gamma_{k+1}}. \tag{27}$$

7. Hybrid modified accelerated gradient descent method (*HMAGD*) [32]

$$x_{k+1} = x_k - (\alpha_k + 1)\theta_k(t_k + t_k^2 - t_k^3)g_k, \quad \alpha \in (0,1) \quad \forall k,\ \theta_k = \frac{\gamma_k}{t_k\gamma_{k+1}}. \tag{28}$$

8. Hybrid modified improved gradient descent method (*HMIGD*) [32]

$$x_{k+1} = x_k - (\alpha_k + 1)\gamma_k^{-1}(t_k + t_k^2 - t_k^3)g_k, \quad \alpha \in (0,1) \quad \forall k. \tag{29}$$

As shown above, from Khan's hybridization rule at least eight hybrid models have appeared. Convergence properties as well as performance efficiency of these iterative schemes are presented and illustrated in the literature. The leading model of all listed ((17), (19), (21), (23), (26), (27), (28), (29)), and therewith the first one that was developed on the basis of the Khan's process, is certainly the *HSM* method from [28]. In paper [28], the authors examined the performance of the defined method for various values of the so-called correction parameter $\alpha_k \in (0,1)$, i.e., $\alpha = \alpha_k + 1 \in (1,2)$, which is a necessary factor of all hybrid methods generated through Khan's hybridization. They experimentally concluded that the *HSM* method achieves best performance characteristics when the correction parameter is taken closely to its left limit. Later in [33], the authors improved the *HSM* model by reducing the initial step length parameter in the Backtracking procedure.

3. Dolan–Moré Performance Profiles and Comparisons

In this section, we apply the aspects of Dolan–Moré benchmarking optimization software from [34] on hybrid and non-hybrid variants of chosen accelerated gradient minimization models. In conducted numerical tests, we follow performance metrics regarding the number of iterations, CPU time and number of function evaluations.

For all obtained numerical outcomes, the following points are valid:

- Codes are written in the visual C++ programming language and run on a Workstation Intel(R) Core(TM) 2.3 GHz.
- The Backtracking parameters values taken are: $\sigma = 0.0001$ and $\beta = 0.8$. These are standard values for the Backtracking parameters applied in various optimization models with Backtracking algorithm [2,14,15,20–22,28–31]. This set of values means that a small portion of the decrease predicted by the linear approximation of the current point is accepted.
- The stopping criteria are:

$$\|g_k\| \leq 10^{-6} \quad \text{and} \quad \frac{|f(x_{k+1}) - f(x_k)|}{1 + |f(x_k)|} \leq 10^{-16}.$$

- Chosen test functions are taken from the unconstrained test functions collection presented in [35]. More precisely, all specific test functions that were used for this research are listed in Listing 1.

Listing 1. Test functions.

1. Extended Penalty
2. Perturbed Quadratic
3. Raydan-1
4. Diagonal 1
5. Diagonal 3
6. Generalized Tridiagonal-1
7. Diagonal 4
8. Extended Himmelblau
9. Quadr. Diag. Perturbed
10. Quadratic QF1
11. Exten. Quadr. Penalty QP1
12. Exten. Quadr. Penalty QP2
13. Quadratic QF2
14. Extended EP1
15. Arwhead
16. Almost Perturbed Quadratic
17. Engval1
18. Quartc
19. Generalized Quartic
20. LIARWHD
21. Diagonal 6
22. Tridia
23. Indef
24. Diagonal 9
25. DIXON3DQ
26. NONSCOMP
27. BIGGSB1
28. Power (Cute)
29. Hager
30. Raydan 2

Further, by $i_{p,s}$, $t_{p,s}$ and $e_{p,s}$ we denote the number of iterations, the CPU time and the number of function evaluations, respectively, needed for solving problem p when the solver s is applied. The main observation arises from a comparison of performance profiles, considering analyzed metrics, of hybrid and non-hybrid versions of the same scheme. For this investigation, we chose the following three accelerated gradient methods: *SM* (8), *ADD* (9) and *TADSS* (14). Accordingly, we analyzed their hybrid forms *HSM* (17), *HADD* (19) and *HTADSS* (23), as well. So in these tests, the solver s belongs to the set of six elements, $s \in \{SM, HSM, ADD, HADD, TADSS, HTADSS\}$. We specifically chose this set of comparative non-hybrid and hybrid pairs among the others mentioned above in Section 2, since the selected models are the most cited of Khan's hybrid methods.

According to the benchmark presented in [34] and regarding the comparisons obtained within this paper, for each pair of comparative hybrid and non-hybrid variants we have two solvers, i.e., $n_s = 2$, where the comparative pairs are:

$$\{(SM, HSM), (ADD, HADD), (TADSS, HTADSS)\}.$$

The total number of experiments for each pair is minimal $n_p = 210$. Precisely, for the pair (SM, HSM) we have conducted numerical tests for 25 test function and 11 different numbers of variables, so $n_p = 11 \cdot 25 = 275$, the same as for the pair $(TADSS, HTADSS)$. The pair $(ADD, HADD)$ included 21 test functions with 10 different numbers of variables, so in this case $n_p = 210$. In order to apply Dolan–Moré benchmarking optimization software with performance profiles over the chosen accelerated and hybrid models, we use the original outcomes presented in the papers in which these models were generated [28,29,31]. So, since in [28,31] numerical experiments included 25 test functions, while in [29] the number of tested functions is 21, the total number of tests for all test functions and all three pairs of models is $n_p^{SM,HSM} + n_p^{TADSS,HTADSS} + n_p^{ADD,HADD} = 760$.

Considering defined parameters, we are now able to expose performance ratios defined for the number of iterations, the CPU time and the number of function evaluations, respectively:

$$\begin{aligned}
r_{p,s} &= \frac{i_{p,s}}{\min\{i_{p,s} : s \in \{SM, HSM\}\}} \\
&= \frac{i_{p,s}}{\min\{i_{p,s} : s \in \{ADD, HADD\}\}} = \frac{i_{p,s}}{\min\{i_{p,s} : s \in \{TADSS, HTADSS\}\}}, \\
r_{p,s} &= \frac{t_{p,s}}{\min\{t_{p,s} : s \in \{SM, HSM\}\}} \\
&= \frac{t_{p,s}}{\min\{t_{p,s} : s \in \{ADD, HADD\}\}} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \{TADSS, HTADSS\}\}}, \\
r_{p,s} &= \frac{e_{p,s}}{\min\{e_{p,s} : s \in \{SM, HSM\}\}} \\
&= \frac{e_{p,s}}{\min\{e_{p,s} : s \in \{ADD, HADD\}\}} = \frac{e_{p,s}}{\min\{e_{p,s} : s \in \{TADSS, HTADSS\}\}}.
\end{aligned}$$

As in [34], we define the performance profile for each solver s with respect to all three measured metrics

$$\rho_s(\tau) = \frac{1}{n_p} size\{p \in \mathcal{P} : r_{p,s} \leq \tau\}, \tag{30}$$

which presents the cumulative distribution function. In (30) parameter $\tau \in \mathbb{R}$, while $\mathcal{P}$ is the set of problems.

In Figures 1–3, we present the performance profiles of SM and HSM regarding the number of iterations, the CPU time and the number of function evaluations, respectively. Comparisons between pairs ($ADD, HADD$) and ($TADSS, HTADSS$), regarding all three tested metrics are similarly illustrated in Figures 4–9.

From Figures 1–3, we clearly observe that the HSM algorithm outperforms the non-hybrid SM model, with respect to the number of iterations and the CPU time metrics, while regarding the number of evaluations metric, the hybrid and non-hybrid models perform similarly.

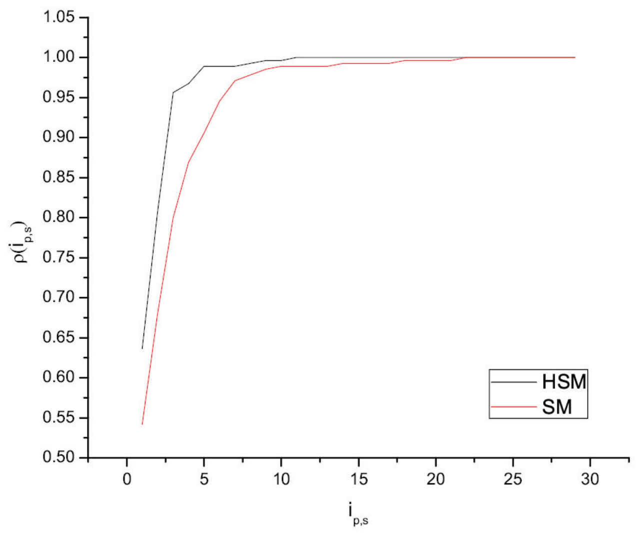

Figure 1. Performance profiles for the HSM and SM methods regarding the number of iterations metric.

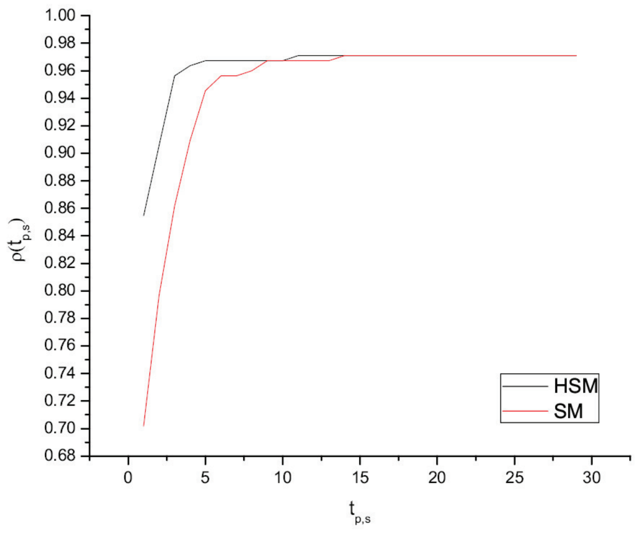

Figure 2. Performance profiles for the HSM and SM methods regarding the CPU time metric.

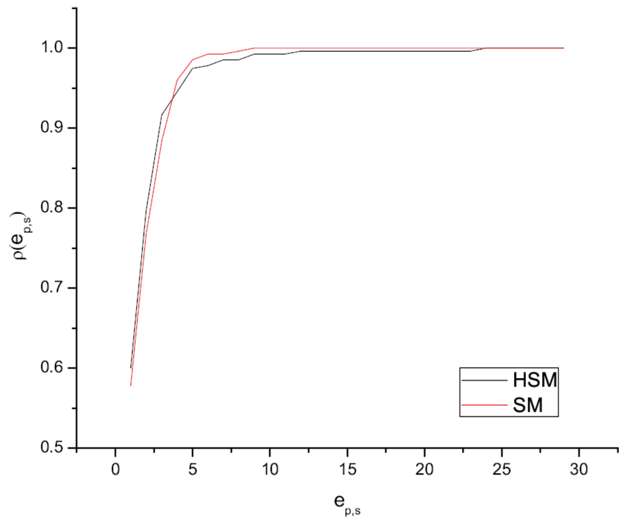

Figure 3. Performance profiles for the HSM and SM methods regarding the number of function evaluations metric.

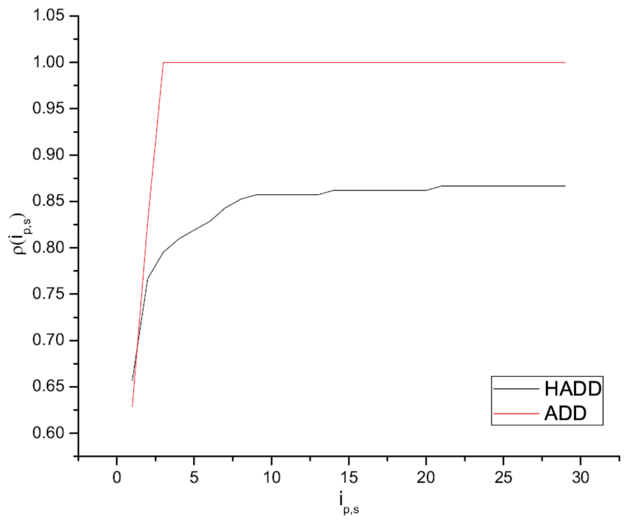

Figure 4. Performance profiles for the HADD and ADD methods regarding the number of iterations metric.

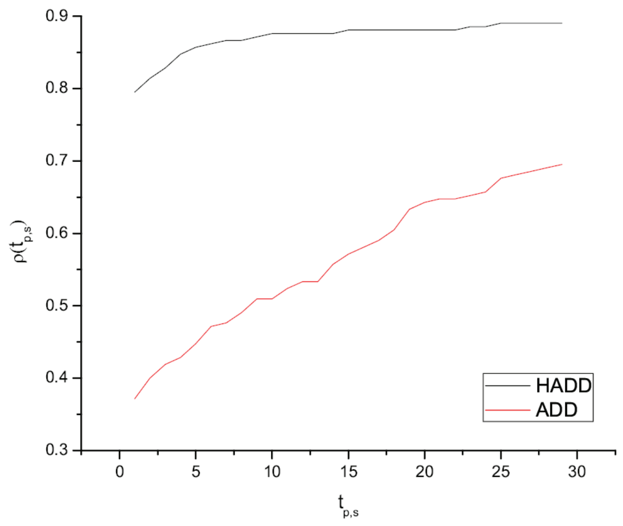

Figure 5. Performance profiles for the HADD and ADD methods regarding the CPU time metric.

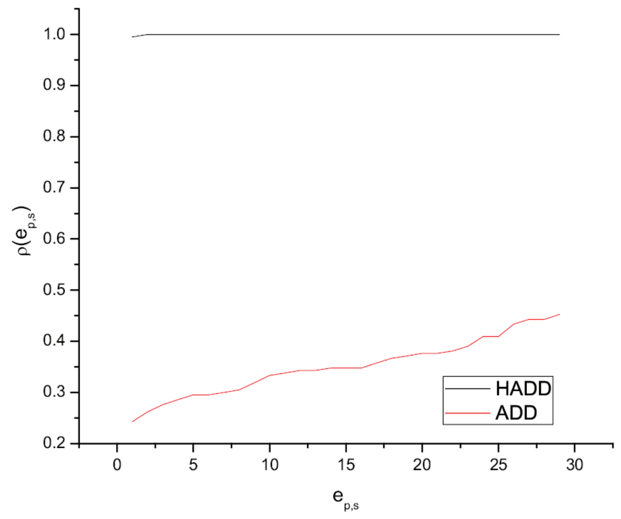

Figure 6. Performance profiles for the HADD and ADD methods regarding the number of function evaluations metric.

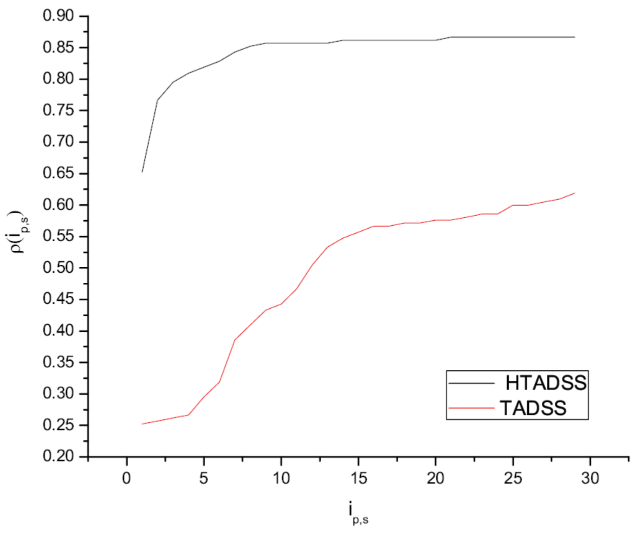

Figure 7. Performance profiles for the HTADSS and TADSS methods regarding the number of iterations metrics.

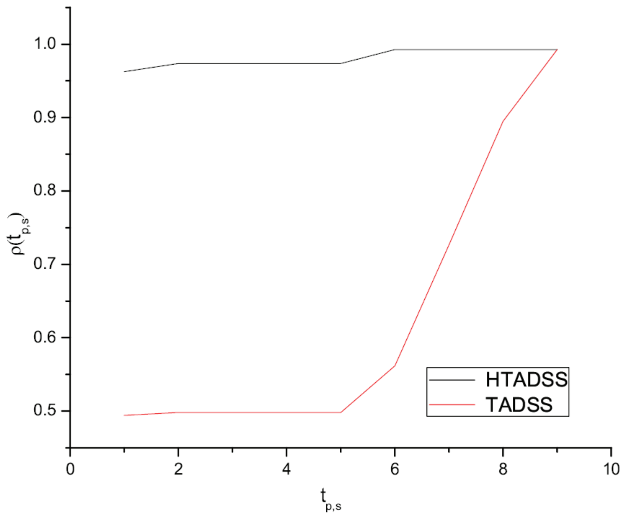

Figure 8. Performance profiles for the HTADSS and TADSS methods regarding the CPU time metric.

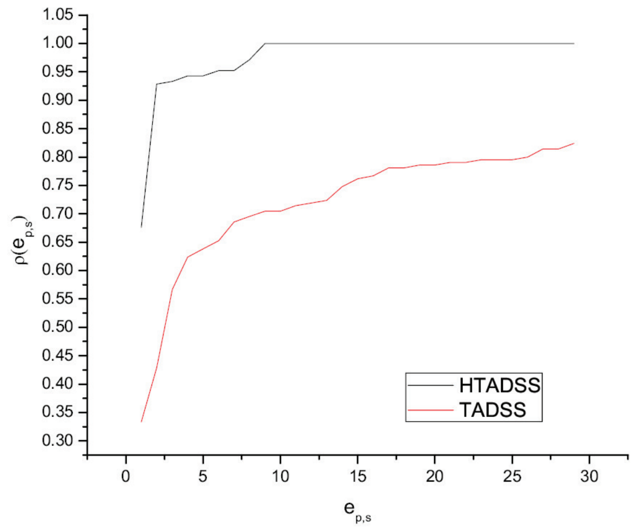

Figure 9. Performance profiles for the HTADSS and TADSS methods regarding the number of function evaluations metric.

From Figures 5 and 6, we see that the hybrid accelerated double direction model shows conspicuously better features regarding CPU time and the number of evaluations metrics. Nevertheless, concerning the number of iterations metric, forerunner ADD is more efficient, as shown in Figure 4.

Finally, comparisons of the $TADSS$ and the $HTADSS$ methods are displayed in Figures 7–9. From these three presented graphs, we observe that the hybrid version of the transformed double step size method convincingly upgrades its counterpart non-hybrid model. In this case, the dominance of the hybrid variant with respect to all three analyzed metrics is more than evident.

To obtain Figures 1–9, a total of 4560 numerical outcomes were included. More precisely, for 6 analyzed methods (SM, ADD, $TADSS$, HSM, $HADD$, $HTADSS$) we followed 3 metrics (number of iterations, CPU time, number of function evaluations) on 25 test functions for SM, HSM, $TADSS$ and $HTADSS$ solvers and 21 test functions for ADD and $HADD$ solvers from [35]. For each test function, the tests were conducted for at least 10 different numbers of variables. With that, the execution time for each test is limited by the time-limiter parameter defined in [16].

4. Conclusions

In this research, we present an overview of two gradient method classes: accelerated gradient descent models and its hybrid variants derived from Khan's three-term iterative rule. This is an important and useful retrospective of one, confirmed efficient approach in defining the robust accelerated methods for solving unconstrained optimization problems. The obtained results, achieved on the basis of comprehensive Dolan–Moré performance profiles [34], conducted on total 4560 numerical outcomes, confirm that Khan's hybridization rule is justified for use as an applicable technique in generating effective minimization processes. Accordingly, this research paves the way for new possibilities aimed at generating similar hybridization rules and their applications to accelerated gradient schemes.

Author Contributions: Conceptualization, V.R. and M.J.P.; Methodology, V.R.; Software, M.J.P.; Validation, V.R.; Formal analysis, V.R.; Investigation, M.J.P.; Resources, V.R.; Data curation, M.J.P.; Writing–original draft, M.J.P.; Writing–review & editing, V.R.; Visualization, M.J.P.; Supervision, V.R.; Project administration, M.J.P. All authors contributed equally and significantly to the writing of this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by internal-junior project IJ-0202, Faculty of Sciences and Mathematics, University of Priština in Kosovska Mitrovica.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data results are available on request from the corresponding author.

Acknowledgments: The authors gratefully acknowledge support from the project Grant No. 174025 by the Ministry of Education and Science of Republic of Serbia.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Powel, M.J.D. A survey of numerical methods for unconstrained optimization. *SIAM Rev.* **1970**, *12*, 79–97. [CrossRef]
2. Andrei, N. Nonlinear Conjugate Gradient Methods for Unconstrained Optimization. In *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*; Springer: Berlin/Heidelberg, Germany, 2020.
3. Nocadal, J.; Wright, S.J. Numerical Optimization. In *Numerical Optimization*; Springer: New York, NY, USA, 1999.
4. Jacoby, S.L.S.; Kowalik, J.S.; Pizzo, J.T. Iterative Methods for Nonlinear Optimization Problems. In *Iterative Methods for Nonlinear Optimization Problems*; Prentice-Hall, Inc.: Englewood, NJ, USA, 1977.
5. Deniss, J.E.; Kowalik, J.S.; Schnabel, R.B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. In *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1983.

6. Fletcher, R. Practical methods of optimization. In *Practical Methods of Optimization*; Prentice-Hall, Wiley: New York, NY, USA, 2000.
7. Luenberg, D.G.; Ye, Y. Linear and nonlinear programming. In *Linear and Nonlinear Programming*; Springer Science + Business Media, LLC: Berlin/Heidelberg, Germany, 2008.
8. Sun, W.; Yuan, Y.X. Optimization Theory and Methods: Nonlinear Programming. In *Optimization Theory and Methods: Nonlinear Programming*; Springer: New York, NY, USA, 2006.
9. Shi, Z.J. Convergence of line search methods for unconstrained optimization. *Appl. Math. Comput.* **2004**, *151*, 393–405. [CrossRef]
10. Wolfe, P. Convergence conditions for ascent methods. *SIAM Rev.* **1968**, *11*, 226–235. [CrossRef]
11. Ortega, J.M.; Rheinboldt, W.C. Iterative Solution of Nonlinear Equation in Several Variables. In *Iterative Solution of Nonlinear Equation in Several Variables*; Academic Press: Cambridge, MA, USA, 1970.
12. Armijo, L. Minimization of functions having Lipschitz continuous first partial derivatives. *Pac. J. Math.* **2008**, *16*, 1–3. [CrossRef]
13. Brezinski, C. A classification of quasi-Newton methods. *Numer. Algor.* **2003**, *33*, 123–135. [CrossRef]
14. Stanimirovic, P.S.; Miladinović, M.B. Accelerated gradient descent methods with line search. *Numer. Algor.* **2010**, *54*, 503–520. [CrossRef]
15. Andrei, N. An acceleration of gradient descent algoritham with backtracing for unconstrained optimization. *Numer. Algor.* **2006**, *42*, 63–173. [CrossRef]
16. Petrović, M.J.; Stanimirovic, P.S. Accelerated Double Direction Method For Solving Unconstrained Optimization Problems. *Math. Probl. Eng.* **2014**, *2014*, 965104. [CrossRef]
17. Petrović, M.J. An accelerated Double Step Size method in unconstrained optimization. *Appl. Math. Comput.* **2015**, *250*, 309–319. [CrossRef]
18. Stanimirovic, P.S.; Petrović, M.J.; Milovanović, G.V. A Transformation of Accelerated Double Step Size Method for Unconstrained Optimization. *Math. Probl. Eng.* **2015**, *2015*, 283679. [CrossRef]
19. Petrović, M.J.; Valjarević, D.; Ilić, D.; Valjarević, A.; Mladenović, J. An improved modification of accelerated double direction and double step-size optimization schemes. *Mathematics* **2022**, *10*, 259. [CrossRef]
20. Barzilai, B.; Borwein, J.M. Two point step-size gradient method. *IMA J. Numer. Anal.* **1988**, *8*, 141–148. [CrossRef]
21. Miladinović, M.; Stanimirović, P.S.; Miljković, S. Scalar correction method for solving large scale' unconstrained minimization problems. *J. Optim. Theory Appl.* **2011**, *151*, 304–320. [CrossRef]
22. Andrei, N. A new three-term conjugate gradient algorithm for unconstrained optimization. *Numer. Algor.* **2014**, *68*, 305–321. [CrossRef]
23. Djuranović-Miličić, N.I.; Gardašević-Filipović, M. A multi-step curve search algorithm in nonlinear optimization: Nondifferentiable convex case. *Facta Univ. Ser. Math. Inform.* **2010**, *25*, 11–24.
24. Picard, E. Memoire sur la theorie des equations aux derivees partielles et la methode des approximations successives. *J. Math. Pures Appl.* **1890**, *6*, 145–210.
25. Mann, W.R. Mean value methods in iterations. *Proc. Am. Math. Soc.* **1953**, *4*, 506–510. [CrossRef]
26. Ishikawa, S. Fixed points by a new iteration method. *Proc. Am. Math. Soc.* **1974**, *44*, 147–150. [CrossRef]
27. Khan, S.H. A Picard-Mann hybrid iterative process. *Fixed Point Theory Appl.* **2013**, *2013*, 69. [CrossRef]
28. Petrović, M.J.; Rakočević, V.; Kontrec, N.; Panić, S.; Ilić, D. Hybridization Accel. Gradient Descent Method. *Numer. Algor.* **2018**, *79*, 769–786. [CrossRef]
29. Petrović, M.J.; Stanimirović, P.S.; Kontrec, N.; Maldenović, J. Hybrid modification of accelerated double direction method. *Math. Probl. Eng.* **2018**, *2018*, 1523267. [CrossRef]
30. Petrović, M.J. Hybridization Rule Applied on Accelerated Double Step Size Optimization Scheme. *Filomat* **2019**, *33*, 655–665. [CrossRef]
31. Petrović, M.J.; Rakočević, V.; Valjarević, D.; Ilić, D. A note on hybridization process applied on transformed double step size model. *Numer. Algor.* **2020**, *85*, 449–465. [CrossRef]
32. Ivanov, M.J., Stanimirović, P.S., Milovanović, G.V.; Djordjević, S.; Brajević, I. Accelerated Multi Step-Size Methods Solving Unconstrained Optimization Problems. *Optim. Method Softw.* **2020**, *85*, 449–465. [CrossRef]
33. Petrović, M.J.; Panić, S.; Carerević, M.M. Initial improvement of the hybrid accelerated gradient descent process. *Bull. Aust. Math. Soc.* **2018**, *98*, 331–338. [CrossRef]
34. Dolan, E.; Moré, J. Benchmarking optimization software with performance profiles. *Math. Program* **2002**, *91*, 201–213. [CrossRef]
35. Andrei, N. An Unconstrained Optimization Test Functions Collection. *Adv. Model. Optim.* **2008**, *10*, 147–161. Available online: https://camo.ici.ro/journal/vol10/v10a10.pdf (accessed on 16 November 2022).