Chapter

Some Unconstrained Optimization Methods

Snezana S. Djordjevic

Abstract

Although it is a very old theme, unconstrained optimization is an area which is always actual for many scientists. Today, the results of unconstrained optimization are applied in different branches of science, as well as generally in practice. Here, we present the line search techniques. Further, in this chapter we consider some unconstrained optimization methods. We try to present these methods but also to present some contemporary results in this area.

Keywords: unconstrained optimization, line search, steepest descent method, Barzilai-Borwein method, Newton method, modified Newton method, inexact Newton method, quasi-Newton method

1. Introduction

Optimization is a very old subject of a great interest; we can search deep into a human history to find important examples of applying optimization in the usual life of a human being, for example, the need of finding the best way to produce food yielded finding the best piece of land for producing, as well as (later on, how the time was going) the best ways of treatment of the chosen land and the chosen seedlings to get the best results.

From the very beginning of manufacturing, the manufacturers were trying to find the ways to get maximum income with minimum expenses.

There are plenty of examples of optimization processes in pharmacology (for determination of the geometry of a molecule), in meteorology, in optimization of a trajectory of a deep-water vehicle, in optimization of power management (optimization of the production of electrical power plants), etc.

Optimization presents an important tool in decision theory and analysis of physical systems.

Optimization theory is a very developed area with its wide application in science, engineering, business management, military, and space technology.

Optimization can be defined as the process of finding the best solution to a problem in a certain sense and under certain conditions.

Along with the passage of time, optimization was evolving. Optimization became an independent area of mathematics in 1940, when Dantzig presented the so-called simplex algorithm for linear programming.

The development of nonlinear programming became great after presentation of conjugate gradient methods and quasi-Newton methods in the 1950s.

Today, there exist many modern optimization methods which are made to solve a variety of optimization problems. Now, they present the necessary tool for solving problems in diverse fields.

At the beginning, it is necessary to define an objective function, which, for example, could be a technical expense, profit or purity of materials, time, potential energy, etc.

The object function depends on certain characteristics of the system, which are known as variables. The goal is to find the values of those variables, for which the object function reaches its best value, which we call an extremum or an optimum.

It can happen that those variables are chosen in such a way that they satisfy certain conditions, i.e., restrictions.

The process of identifying the object function, variables, and restrictions for the given problem is called *modeling*.

The first and the most important step in an optimization process is the construction of the appropriate model, and this step can be the problem by itself. Namely, in the case that the model is too much simplified, it cannot be a faithful reflection of the practical problem. By the other side, if the constructed model is too complicated, then solving the problem is also too complicated.

After the construction of the appropriate model, it is necessary to apply the appropriate algorithm to solve the problem. It is no need to emphasize that there does not exist a universal algorithm for solving the set problem.

Sometimes, in the applications, the set of input parameters is bounded, i.e., the input parameters have values within the allowed space of input parameters D_x; we can write

$$x \in D_x. \tag{1}$$

Except (1), the next conditions can also be imposed:

$$\varphi_l(x_1, ..., x_n) = \varphi_{0l}, l = 1, ..., m_1 < n, \tag{2}$$

$$\psi_j(x_1, ..., x_n) \leq \psi_{0j}, j = 1, ..., m_2. \tag{3}$$

Optimization task is to find the minimum (maximum) of the objective function $f(x) = f(x_1, ..., x_n)$, under the conditions (1), (2), and (3).

If the object function is linear, and the functions $\varphi_l(x_1, ..., x_n)\, l = 1, ..., m_1$ and $\psi_j(x_1, ..., x_n) j = 1, ..., m_2$ are linear, then it is about the linear programming problem, but if at least one of the mentioned functions is nonlinear, it is about the nonlinear programming problem.

Unconstrained optimization problem can be presented as

$$\min_{x \in R^n} f(x), \tag{4}$$

where $f \in R^n$ is a smooth function.

Problem (4) is, in fact, the unconstrained minimization problem. But, it is well known that the unconstrained minimization problem is equivalent to an unconstrained maximization problem, i.e.

$$\min f(x) = -\max(-f(x)), \tag{5}$$

as well as

$$\max f(x) = -\min(-f(x)). \tag{6}$$

Definition 1.1.1 *x^* is called a global minimizer of f if $f(x^*) \leq f(x)$ for all $x \in R^n$.*

The ideal situation is finding a global minimizer of f. Because of the fact that our knowledge of the function f is usually only local, the global minimizer can be very difficult to find. We usually do not have the total knowledge about f. In fact, most algorithms are able to find only a local minimizer, i.e., a point that achieves the smallest value of f in its neighborhood.

So, we could be satisfied by finding the local minimizer of the function f. We distinguish weak and strict (or strong) local minimizer.

Formal definitions of local weak and strict minimizer of the function f are the next two definitions, respectively.

Definition 1.1.2 *x^* is called a weak local minimizer of f if there exists a neighborhood N of x^*, such that $f(x^*) \leq f(x)$ for all $x \in N$.*

Definition 1.1.3 *x^* is called a strict (strong) local minimizer of f if there exists a neighborhood N of x^*, such that $f(x^*) < f(x)$ for all $x \in N$.*

Considering backward definitions 1.1.2 and 1.1.3, the procedure of finding local minimizer (weak or strict) does not seem such easy; it seems that we should examine all points from the neighborhood of x^*, and it looks like a very difficult task.

Fortunately, if the object function f satisfies some special conditions, we can solve this task in a much easier way.

For example, we can assume that the object function f is smooth or, furthermore, twice continuously differentiable. Then, we concentrate to the gradient $\nabla f(x^*)$ as well as to the Hessian $\nabla^2 f(x^*)$.

All algorithms for unconstrained minimization require the user to start from a certain point, so-called the starting point, which we usually denote by x_0. It is good to choose x_0 such that it is a reasonable estimation of the solution. But, to find such estimation, a little more knowledge about the considered set of data is needed, and the systematic investigation is needed also. So, it seems much simpler to use one of the algorithms to find x_0 or to take it arbitrarily.

There exist two important classes of iterative methods—*line search methods* and *trust-region methods*—made in the aim to solve the unconstrained optimization problem (4).

In this chapter, at first, we discuss different kinds of line search. Then, we consider some line search optimization methods in details, i.e., we study steepest descent method, Barzilai-Borwein gradient method, Newton method, and quasi-Newton method.

Also, we try to give some of the most recent results in these areas.

2. Line search

Now, let us consider the problem

$$\min_{x \in R^n} f(x), \tag{7}$$

where $f : R^n \to R$ is a continuously differentiable function, bounded from below.

There exists a great number of methods made in the aim to solve the problem (7).

The optimization methods based on line search utilize the next iterative scheme:

$$x_{k+1} = x_k + t_k d_k, \tag{8}$$

where x_k is the current iterative point, x_{k+1} is the next iterative point, d_k is the search direction, and t_k is the step size in the direction d_k.

At first, we consider the monotone line search.

Now, we give the iterative scheme of this kind of search.

Algorithm 1.2.1. *(Monotone line search).*

Assumptions: $\epsilon > 0$, x_0, $k := 0$.

Step 1. If $\|g_k\| \leq \epsilon$, then STOP.

Step 2. Find the descent direction d_k.

Step 3. Find the step size t_k, such that $f(x_k + t_k d_k) < f(x_k)$.

Step 4. Set $x_{k+1} = x_k + t_k d_k$.

Step 5. Take $k := k + 1$ and go to Step 1.

Denote

$$\Phi(t) = f(x_k + t d_k).$$

Trying to solve the minimization problem, we are going to search for the step size $t = t_k$, in the direction d_k, such that the next relation holds:

$$\Phi(t_k) < \Phi(0).$$

That procedure is called the monotone line search.

We can search for the step size t_k in such a way that the next relation holds:

$$f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + t_k d_k), \tag{9}$$

i.e.

$$\Phi(t_k) = \min_{t \geq 0} \Phi(t), \tag{10}$$

or we can use the next formula:

$$t_k = \min\left\{t | g(x_k + t d_k)^T d_k = 0, t \geq 0\right\}. \tag{11}$$

In this case we are talking about *the exact* or *the optimal* line search, where the parameter t_k, which is received as the solution of the one-dimensional problem (10), is *the optimal step size*.

By the other side, instead of using the relation (9), or the relation (11), we can be satisfied by searching for such t_k, which is acceptable if the next relation suits us:

$$f(x_k) - f(x_k + t_k d_k) > \delta_k > 0.$$

Then, we are talking about *the inexact* or *the approximate* or *the acceptable* line search, which is very much utilized in the practice.

There are several reasons to use the inexact instead of the exact line search. One of them is that the exact line search is expensive. Further, in the cases when the iteration is far from the solution, the exact line search is not efficient. Next, in the practice, the convergence rate of many optimization methods (such as Newton or quasi-Newton) does not depend on the exact line search.

First, we are going to mention so-called basic and, by the way, very well-known inexact line searches.

Algorithm 1.2.2. *(Backtracking).*

Assumptions: x_k, the descent direction d_k, $0 < \delta < \frac{1}{2}$, $\eta \in (0,1)$.

Step 1. $t := 1$.

Step 2. While $f(x_k + td_k) > f(x_k) + \delta t g_k^T d_k$, $t := t \cdot \eta$.
Step 3. Set $t_k = t$.
Now, we describe the Armijo rule.

Theorem 1.2.1. [1] *Let $f \in C^1(R^n)$ and let d_k be the descent direction. Then, there exists the nonnegative number m_k, such that*

$$f(x_k + \eta^{m_k} d_k) \leq f(x_k) + c_1 \eta^{m_k} g_k^T d_k,$$

where $c_1 \in (0,1)$ and $\eta \in (0,1)$.
Next, we describe the Goldstein rule [2].
The step size t_k is chosen in such a way that

$$f(x_k + td_k) \leq f(x_k) + \delta t g_k^T d_k,$$
$$f(x_k + td_k) > f(x_k) + (1-\delta) t g_k^T d_k,$$

where $0 < \delta < \frac{1}{2}$.
Now, Wolfe line search rules follow [3], [4].
Standard Wolfe line search conditions are

$$f(x_k + t_k d_k) - f(x_k) \leq \delta t_k g_k^T d_k, \tag{12}$$
$$g_{k+1}^T d_k \geq \sigma g_k^T d_k, \tag{13}$$

where d_k is a descent direction and $0 < \delta \leq \sigma < 1$.
This efficient strategy means that we should accept a positive step length t_k, if conditions (12)–(13) are satisfied.
Strong Wolfe line search conditions consist of (12) and the next, stronger version of (13):

$$|g_{k+1}^T d_k| \leq -\sigma g_k^T d_k. \tag{14}$$

In the generalized Wolfe line search conditions, the absolute value in (14) is replaced by the inequalities:

$$\sigma_1 g_k^T d_k \leq g_{k+1}^T d_k \leq -\sigma_2 g_k^T d_k,\ 0 < \delta \leq \sigma_1 < 1,\ \sigma_2 \geq 0. \tag{15}$$

By the other side, in the approximate Wolfe line search conditions, the inequalities (15) are changed into the next ones:

$$\sigma g_k^T d_k \leq g_{k+1}^T d_k \leq (2\delta - 1) g_k^T d_k,\ 0 < \delta < \frac{1}{2},\ \delta < \sigma < 1. \tag{16}$$

The next lemma is very important.

Lemma 1.2.1. [5] *Let $f \in C(R^n)$. Let d_k be a descent direction at the point x_k, and assume that the function f is bounded from below along the direction $\{x_k + td_k | t > 0\}$. Then, if $0 < \delta < \sigma < 1$, there exist the intervals inside which the step length satisfies standard Wolfe conditions and strong Wolfe conditions.*

By the other side, the introduction of the non-monotone line search is motivated by the existence of the problems where the search direction does not have to be a descent direction. This can happen, for example, in stochastic optimization [6].

Next, some efficient quasi-Newton methods, for example, *SR*1 update, do not produce the descent direction in every iteration [5].

Further, some efficient methods like spectral are not monotone at all.

Some numerical results given in [7–11] show that non-monotone techniques are better than the monotone ones if the problem is to find the global optimal values of the object function.

Algorithms of the non-monotone line search do not insist on a descent of the object function in every step. But, even these algorithms require the reduction of the object function after a predetermined number of iterations.

The first non-monotone line search technique is presented in [12]. Namely, in [12], the problem is to find the step size which satisfies

$$f(x_k + t_k d_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \delta t_k g_k^T d_k,$$

where $m(0) = 0$, $0 \leq m(k) \leq \min\{m(k-1)+1, M\}$, for $k \geq 1$, $\delta \in (0,1)$, where M is a nonnegative integer.

This strategy is in fact the generalization of Armijo line search. In the same work, the authors suppose that the search directions satisfy the next conditions for some positive constants b_1 and b_2:

$$g_k^T d_k \leq -b_1 \|g_k\|^2,$$
$$\|d_k\| \leq b_2 \|g_k\|.$$

The next non-monotone line search is described in [11].

Let x_0 be the starting point, and let

$$0 \leq \eta_{min} \leq \eta_{max} \leq 1,\ 0 < \delta < \sigma < 1 < \rho,\ \mu > 0.$$

Let $C_0 = f(x_0)$, $Q_0 = 1$.

The step size has to satisfy the next conditions:

$$f(x_k + t_k d_k) \leq C_k + \delta t_k g_k^T d_k, \tag{17}$$

$$g(x_k + t_k d_k) \geq \sigma g_k^T d_k. \tag{18}$$

The value η_k is chosen from the interval $[\eta_{min}, \eta_{max}]$ and then

$$Q_{k+1} = \eta_k Q_k + 1,\ C_{k+1} = \frac{\eta_k Q_k C_k + f(x_{k+1})}{Q_{k+1}}.$$

Non-monotone rules which contain the sequence of nonnegative parameters $\{\epsilon_k\}$ are used firstly in [13], and they are successfully used in many other algorithms, for example, in [14]. The next property of the parameters ϵ_k is assumed:

$$\epsilon_k > 0, \quad \sum_k \epsilon_k = \epsilon < \infty,$$

and the corresponding rule is

$$f(x_k + t_k d_k) \leq f(x_k) + c_1 t_k g_k^T d_k + \epsilon_k.$$

Now, we give the non-monotone line search algorithm, shortly *NLSA*, presented in [11].

Algorithm 1.2.3. *(NLSA).*

Assumptions: x_0, $0 \leq \eta_{min} \leq \eta_{max} \leq 1$, $0 < \delta < \sigma < 1 < \rho$, $\mu > 0$.

Set $C_0 = f(x_0)$, $Q_0 = 1$, $k = 0$.

Step 1. If $\|\nabla f(x_k)\|$ is sufficiently small, then STOP.

Step 2. Set $x_{k+1} = x_k + t_k d_k$, where t_k satisfies either the (non-monotone) Wolfe conditions (17) and (18) or the (non-monotone) Armijo conditions: $t_k = \bar{t}_k \rho^{h_k}$, where $\bar{t}_k > 0$ is the trial step and h_k is the largest integer such that (17) holds and $t_k \leq \mu$.

Step 3. Choose $\eta_k \in [\eta_{min}, \eta_{max}]$, and set

$$Q_{k+1} = \eta_k Q_k + 1, C_{k+1} = (\eta_k Q_k C_k + f(x_{k+1}))/Q_{k+1}.$$

Step 4. Set $k := k + 1$ and go to Step 1.

We can notice [11] that C_{k+1} is a convex combination of $f(x_0), f(x_1), \ldots, f(x_k)$. The parameter η_k controls the degree of non-monotonicity.

If $\eta_k = 0$ for all k, then this non-monotone line search becomes monotone Wolfe or Armijo line search.

If $\eta_k = 1$ for all k, then $C_k = A_k$, where

$$A_k = \frac{1}{k+1} \sum_{i=0}^{k} f(x_i).$$

Lemma 1.2.2. [11] *If $\nabla f(x_k)^T d_k \leq 0$ for each k, then for the iterates generated by the non-monotone line search algorithm, we have $f_k \leq C_k \leq A_k$ for each k. Moreover, if $\nabla f(x_k)^T d_k < 0$ and $f(x)$ are bounded from below, then there exists t_k satisfying either Wolfe or Armijo conditions of the line search update.*

This study would be very incomplete unless we mention that there are many modifications of the abovementioned line searches. All these modifications are made to improve the previous results.

For example, in [15], the new inexact line search is described by the next way.

Let $\beta \in (0,1)$, $\sigma \in \left(0, \frac{1}{2}\right)$; let B_k be a symmetric positive definite matrix which approximates $\nabla^2 f(x_k)$ and $s_k = -\frac{g_k^T d_k}{d_k^T B_k d_k}$. The step size t_k is the largest one in $\{s_k, s_k\beta, s_k\beta^2, \ldots\}$ such that

$$f(x_k + td_k) - f(x_k) \leq \sigma t \left[g_k^T d_k + \frac{1}{2} t d_k^T B_k d_k \right].$$

Further, in [16], a new inexact line search rule is presented. This rule is a modified version of the classical Armijo line search rule. We describe it now.

Let $g = \nabla f(x)$ be a Lipschitz continuous function and L the Lipschitz constant. Let L_k be an approximation of L. Set

$$\beta_k = -\frac{g_k^T d_k}{L_k \|d_k\|^2}.$$

Find a step size t_k as the largest component in the set $\{\beta_k, \beta_k\rho, \beta_k\rho^2 \ldots\}$ such that the inequality

$$f(x_k + t_k d_k) \leq f(x_k) + \sigma t_k \left(g_k^T d_k - \frac{1}{2} t_k \mu L_k \|d_k\|^2 \right)$$

holds, where $\sigma \in (0,1)$, $\mu \in [0, \infty)$, and $\rho \in (0,1)$ are given constants.

Next, in [17], a new, modified Wolfe line search is given in the next way.

Find $t_k > 0$ such that

$$f(x_k + t_k d_k) - f(x_k) \leq \min\{\delta t_k g_k^T d_k, -\gamma t_k^2 \|d_k\|^2\},$$
$$g(x_k + t_k d_k)^T d_k \geq \sigma g_k^T d_k,$$

where $\delta \in (0,1)$, $\sigma \in (\delta, 1)$, and $\gamma > 0$.

More recent results on this topic can be found, for example, in [18–23].

2.1 Steepest descent (*SD*)

The classical steepest descent method which is designed by Cauchy [24] can be considered as one among the most important procedures for minimization of real-valued function defined on $\mathbb{R}^n$.

Steepest descent is one of the simplest minimization methods for unconstrained optimization. Since it uses the negative gradient as its search direction, it is known also as the gradient method.

It has low computational cost and low matrix storage requirement, because it does not need the computations of the second derivatives to be solved to calculate the search direction [25].

Suppose that $f(x)$ is continuously differentiable in a certain neighborhood of a point x_k and also suppose that $g_k \triangleq \nabla f(x_k) \neq 0$.

Using Taylor expansion of the function f near x_k as well as Cauchy-Schwartz inequality, one can easily prove that the greatest fall of f exists if and only if $d_k = -g_k$, i.e., $-g_k$ is the steepest descent direction.

The iterative scheme of the *SD* method is

$$x_{k+1} = x_k - t_k g_k. \tag{19}$$

The classical steepest descent method uses the exact line search.

Now, we give the algorithm of the steepest descent method which refers to the exact as well as to the inexact line search.

Algorithm 1.2.4. *(Steepest descent method, i.e., SD method).*
Assumptions: $0 < \epsilon \ll 1$, $x_0 \in \mathbb{R}^n$. Let $k = 0$.
Step 1. If $\|g_k\| \leq \varepsilon$, then STOP, else set $d_k = -g_k$.
Step 2. Find the step size t_k, which is the solution of the problem

$$\min_{t \geq 0} f(x_k + t d_k), \tag{20}$$

else find the step size t_k by any of the inexact line search methods.
Step 3. Set $x_{k+1} = x_k + t_k d_k$.
Step 4. Set $k := k + 1$ and go to Step 1.

The classical and the oldest steepest descent step size t_k, which was designed by Cauchy (in the case of the exact line search), is computed as [26]

$$t_k = \frac{g_k^T g_k}{g_k^T G g_k},$$

where $g_k = \nabla f(x_k)$ and $G = \nabla^2 f(x_k)$.

Theorem 1.2.2. [27] *(Global convergence theorem of the SD method) Let $f \in C^1$. Then, each accumulation point of the iterative sequence $\{x_k\}$, generated by Algorithm 1.2.4, is a stationary point.*

Remark 1.2.1. *The steepest descent method has at least the linear convergence rate.*

More information about the convergence of the *SD* method can be found in [5, 27].

Although known as the first unconstrained optimization method, this method is still a theme considered by scientists.

Different modifications of this method are made, for example, see [25, 28–32].

In [28], the authors presented a new search direction from Cauchy's method in the form of two parameters known as *Zubai'ah-Mustafa-Rivaie-Ismail* method, shortly, *ZMRI* method:

$$d_k = -g_k - \|g_k\| g_{k-1}. \tag{21}$$

So, in [28], a new modification of *SD* method is suggested using a new search direction, d_k, given by (21). The numerical results are presented based on the number of iterations and CPU time. It is shown that this new method is efficient when it is compared to the classical *SD*.

In [25], a new scaled search direction of *SD* method is presented. The inspiration for this new method is the work of Andrei [33], in which the author presents and analyzes a new scaled conjugate gradient algorithm, based on an interpretation of the secant equation and on the inexact Wolfe line search conditions.

The method proposed in [25] is known as *Rashidah-Rivaie-Mamat* (*RRM*) method, and it suggests the direction d_k given by the next relation:

$$d_k = \begin{cases} -g_k, \text{ if } k = 0, \\ -\theta_k g_k - \|g_k\| g_{k-1}, \end{cases} \tag{22}$$

where θ_k is a scaling parameter, $\theta_k = \frac{d_{k-1}^T y_{k-1}}{\|g_{k-1}\|^2}$, $y_{k-1} = g_k - g_{k-1}$.

Further, in [25], a comparison among *RRM*, *ZMRI*, and *SD* methods is made; it is shown that *RRM* method is better than *ZMRI* and *SD* methods.

It is interesting that the exact line search is used in [25].

In [34], the properties of steepest descent method from the literature are reviewed together with advantages and disadvantages of each step size procedure.

Namely, the step size procedures, which are compared in this paper, are:

1. $t_k = \frac{g_k^T g_k}{g_k^T H_k g_k}$: Step size method by Cauchy [24], computed by exact line search (*C* step size).

2. Given $s > 0$, $\beta, \sigma \in (0,1)$, $t_k = \max\{s, s\beta, s\beta^2, \ldots\}$ such that

$$f(x_k + t_k d_k) \leq f(x_k) + \sigma t_k g_k^T d_k - \text{Armijo's line search (A step size)}.$$

3. Given $\beta, \sigma \in (0,1)$, $\tilde{t}_0 = 1$, and $t_k = \beta \tilde{t}_k$ such that

$$f(x_k + t_k d_k) \leq f(x_k) + \sigma t_k g_k^T d_k - \text{Backtracking line search (B step size)}.$$

4. $t_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}$, $(BB1)$, $t_k = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$, $(BB2)$, $s_{k-1} = x_k - x_{k-1}$ $y_{k-1} = g_k - g_{k-1}$, : Barzilai and Borwein's formula. The convergence is R-superlinear.

5. $t_k = \frac{t_{k-1}^2 g_k^T g_k}{2\left(f(x_k + t_k d_k) - f(x_k) + t_{k-1} g_k^T g_k\right)}$: Elimination line search (*EL* step size), which estimates the step size without computation of the Hessian.

The comparison is based on time execution, number of total iteration, total percentage of function, gradient and Hessian evaluation, and the most decreased value of objective function obtained.

From the numerical results, the authors conclude that the A method and $BB1$ method are the best methods among others.

Further, in [34], the general conclusions about the steepest descent method are given:

1. This method is sensitive to the initial point.
2. This method has a descent property, and it is a logical starting procedure for all gradient based methods.
3. x_k approaches the minimizer slowly, in fact in a zigzag way.

In [35], in the aim to achieve fast convergence and the monotone property, a new step size for the steepest descent method is suggested.

In [36], for quadratic positive definite problems, an over-relaxation has been considered. Namely, Raydan and Svaiter [36] proved that the poor behavior of the steepest descent method is due to the optimal Cauchy choice of step size and not to the choice of the search direction. These results are extended in [29] to convex, well-conditioned functions. Further, in [29], it is shown that a simple modification of the step length by means of a random variable uniformly distributed in $(0, 1]$, for the strongly convex functions, represents an improvement of the classical gradient descent algorithm. Namely, in this paper, the idea is to modify the gradient descent method by introducing a relaxation of the following form:

$$x_{k+1} = x_k + \theta_k t_k d_k, \tag{23}$$

where θ_k is the relaxation parameter, a random variable uniformly distributed between 0 and 1.

In the recent years, the steepest descent method has been applied in many branches of science; one can be inspired, for example, by [37–43].

2.2 Barzilai and Borwein gradient method

Remind to the fact that SD method performs poorly, converges linearly, and is badly affected by the ill-conditioning.

Also, remind to the fact that this poor behavior of SD method is due to the optimal choice of the step size and not to the choice of the steepest descent direction $-g_k$.

Barzilai and Borwein presented [44] a two-point step size gradient method, which is well known as BB method.

The step size is derived from a two-point approximation to the secant equation.

Consider the gradient iteration form:

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

It can be rewritten as $x_{k+1} = x_k - D_k g_k$, where $D_k = t_k I$.

To make the matrix D_k having quasi-Newton property, the step size t_k is computed in such a way that we get

$$\min \|s_{k-1} - D_k y_{k-1}\|.$$

This yields that

$$t_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, s_{k-1} = x_k - x_{k-1}, y_{k-1} = g_k - g_{k-1}. \tag{24}$$

But, using symmetry, we may minimize $\|D_k^{-1}s_{k-1} - y_{k-1}\|$, with respect to t_k, and we get:

$$t_k^{BB2} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}}, s_{k-1} = x_k - x_{k-1}, y_{k-1} = g_k - g_{k-1}. \tag{25}$$

Now, we give the algorithm of *BB* method.

Algorithm 1.2.5. *(Barzilai-Borwein gradient method, i.e., BB method).*
Assumptions: $0 < \epsilon \ll 1$, $x_0 \in \mathbb{R}^n$. Let $k = 0$.
Step 1. If $\|g_k\| \leq \epsilon$, then STOP, else set $d_k = -g_k$.
Step 2. If $k = 0$, then find the step size t_0 by the line search, else compute t_k using the formula (24) or (25).
Step 3. Set $x_{k+1} = x_k + t_k d_k$.
Step 4. Set $k := k + 1$ and go to Step 1.
Considering Algorithm 1.2.5, we can conclude that this method does not require any matrix computation or any line search.

The Barzilai-Borwein method is in fact the gradient method, which requires less computational work than *SD* method, and it speeds up the convergence of the gradient method. Barzilai and Borwein proved that *BB* algorithm is $R-$superlinearly convergent for the quadratic case.

In the general non-quadratic case, a globalization strategy based on non-monotone line search is applied in this method.

In this general case, t_k, computed by (24) or (25), may be unacceptably large or small. That is the reason why we assume that there exist the numbers t^l and t^r, such that

$$0 < t^l \leq t_k \leq t^r, \text{for all } k.$$

Using the iteration

$$x_{k+1} = x_k - \frac{1}{t_k} g_k = x_k - \lambda_k g_k, \tag{26}$$

with

$$t_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}, \lambda_k = \frac{1}{t_k},$$
$$s_k = -\frac{1}{t_k} g_k = -\lambda_k g_k,$$

we get

$$t_{k+1} = \frac{s_k^T y_k}{s_k^T s_k} = \frac{-\lambda_k g_k^T y_k}{\lambda_k^2 g_k^T g_k} = -\frac{g_k^T y_k}{\lambda_k g_k^T g_k}.$$

Now, we give the algorithm of the Barzilai-Borwein method with non-monotone line search.

Algorithm 1.2.6. *(BB method with non-monotone line search).*

Assumptions: $0 < \epsilon \ll 1$, $x_0 \in \mathbb{R}^n$, $M \geq 0$ is an integer, $\rho \in (0,1)$, $\delta > 0$, $0 < \sigma_1 < \sigma_2 < 1$, t^l, t^r. Let $k = 0$.

Step 1. If $\|g_k\| \leq \epsilon$, then STOP.

Step 2. If $t_k \leq t^l$, or $t_k \geq t^r$, then set $t_k = \delta$.

Step 3. Set $\lambda = \frac{1}{t_k}$.

Step 4. (non-monotone line search) If

$$f(x_k - \lambda g_k) \leq \max_{0 \leq j \leq \min(k,M)} f(x_{k-j}) - \rho \lambda g_k^T g_k,$$

then set

$$\lambda_k = \lambda, x_{k+1} = x_k - \lambda_k g_k,$$

and go to Step 6.

Step 5. Choose $\sigma \in [\sigma_1, \sigma_2]$, set $\lambda = \sigma\lambda$, and go to Step 4.

Step 6. Set $t_{k+1} = -\frac{g_k^T y_k}{\lambda_k g_k^T g_k}$ and $k := k+1$, and return to Step 1.

Obviously, the above algorithm is globally convergent.

Several authors paid attention to the Barzilai-Borwein method, and they proposed some variants of this method.

In [8], the globally convergent Barzilai-Borwein method is proposed by using non-monotone line search by Grippo et al. [12]. In the same paper, Raydan proves the global convergence of the non-monotone Barzilai-Borwein method.

Further, Grippo and Sciandrone [45] propose another type of the non-monotone Barzilai-Borwein method.

Dai [7] gives the basic analysis of the non-monotone line search strategy.

Moreover, in [46] numerical results are presented, using

$$t_k = \frac{s_{\nu(k)}^T y_{\nu(k)}}{s_{\nu(k)}^T s_{\nu(k)}}. \tag{27}$$

and

$$\nu(k) = M_c \cdot \llcorner \frac{k-1}{M_c} \lrcorner,$$

where for $r \in \mathbb{R}$, $\llcorner r \lrcorner$ denotes the largest integer j such that $j \leq r$ and Mc is a positive integer. The gradient method with (27) is called the cyclic Barzilai-Borwein method. Numerical results in [46] prove that their method performs better than the Barzilai-Borwein method.

Many researchers study the gradient method for minimizing a strictly convex quadratic function, namely,

$$\min f(x) = \frac{1}{2} x^T A x - b^T x, \tag{28}$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix and $b \in \mathbb{R}^n$ is a given vector. For an application of the Barzilai-Borwein method to the problem (28), Raydan [47] establishes global convergence, and Dai and Liao [48] prove $\mathbb{R}$-linear rate of convergence. Friedlander, Martinez, Molina, and Raydan [49] propose a new gradient method with retards, in which t_k is defined by

$$t_k = \frac{g_{\nu(k)}^T A^{\rho(k)+1} g_{\nu(k)}}{g_{\nu(k)}^T A^{\rho(k)} g_{\nu(k)}}, \nu(k) \in \{k, k-1, ..., \max\{0, k-m\}\} \tag{29}$$

and $\rho(k) \in \{q_1, ..., q_m\}$, where m is a positive integer and $q_1, ..., q_m \geq -2$ are integers. In the same paper, they establish its global convergence for problem (28) and prove the Q-superlinear rate of convergence in the special case.

In [50], the authors extend the Barzilai-Borwein method, and they give *extended Barzilai-Borwein method*, which they denote *EBB*. They also establish global and $Q-$superlinear convergence properties of the proposed method for minimizing a strictly convex quadratic function. Furthermore, they discuss an application of their method to general objective functions. In [50], a new step size is proposed by extending (29). Namely, in this paper, following Friedlander et al. [49], a new step size is proposed as follows:

$$t_k = \sum_{i=1}^{l} \phi_i \frac{g_{\nu_{i(k)}}^T A^{\rho_i(k)+1} g_{\nu_i(k)}}{g_{\nu_{i(k)}}^T A^{\rho_i(k)} g_{\nu_i(k)}},$$

$$\phi_i \geq 0, \sum_{i=1}^{n} \phi_i = 1,$$

$$\nu_i(k) \in \{k, k-1, ..., \max\{0, k-m\}\}$$

and

$$\phi_i(k) \in \{q_1, ..., q_m\},$$

where l and m are positive integers and $q_1, ..., q_m$ are integers.

Also, an application of algorithm *EBB* to general unconstrained minimization problems (4) is considered.

Following Raydan [8], the authors [50] further combine the non-monotone line search and algorithm *EBB* to get the algorithm called *NEBB*. They also prove the global convergence of the algorithm *NEBB*, under some classical assumptions.

The Barzilai-Borwein method and its related methods are reviewed by Dai and Yuan [51] and Fletcher [52].

In [53], a new concept of the approximate optimal step size for gradient method is introduced and used to interpret the *BB* method; an efficient gradient method with the approximate optimal step size for unconstrained optimization is presented. The next definition is introduced in [53].

Definition 1.2.1. *Let $\Phi(t)$ be an approximation model of $f(x_k - tg_k)$. A positive constant t^* is called approximate optimal step size associated to $\Phi(t)$ for gradient method, if t^* satisfies*

$$t^* = \arg\min_{t>0} \Phi(t).$$

The approximate optimal step size is different from the steepest descent step size, which will lead to the expensive computational cost. The approximate optimal step size is generally calculated easily, and it can be applied to unconstrained optimization.

Due to the effectiveness of t_k^{BB1} and the fact that $t_k^{BB1} = \arg\min_{t>0} \Phi(t)$, we can naturally ask if more suitable approximation models can be constructed to generate more efficient approximate optimal step-sizes.

This is the purpose of work [53]. Further, if the objective function $f(x)$ is not close to a quadratic function on the line segment between x_{k-1} and x_k, in this paper a conic model is developed to generate the approximate optimal step size if the conic model is suitable to be used. Otherwise, the authors consider two cases:

i. If $s_{k-1}^T y_{k-1} > 0$, the authors construct a new quadratic model, to derive the approximate optimal step size.

ii. If $s_{k-1}^T y_{k-1} \leq 0$, they construct a new quadratic model or two other new approximation models to generate the approximate optimal step size for gradient method. They also analyze the convergence of the proposed method under some suitable conditions. Numerical results show the proposed method is better than the BB method.

In [54], derivative-free iterative scheme that uses the residual vector as search direction for solving large-scale systems of nonlinear monotone equations is presented.

The Barzilai-Borwein method is widely used; some interesting results can be found in [55–57].

2.3 Newton method

The basic idea of Newton method for unconstrained optimization is the iterative usage of the quadratic approximation $q^{(k)}$ to the objective function f at the current iterate x_k and then minimization of such approximation $q^{(k)}$.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable, $x_k \in \mathbb{R}^n$, and let the Hessian $\nabla^2 f(x_k)$ be positive definite.

We model f at the current point x_k by the quadratic approximation $q^{(k)}$:

$$f(x_k + s) \approx q^{(k)}(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s, s = x - x_k.$$

Minimization of $q^{(k)}(s)$ gives the next iterative scheme:

$$x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k),$$

which is known as Newton formula.

Denote $G_k = \nabla^2 f(x_k), g_k = \nabla f(x_k)$.

Then, we have a simpler form:

$$x_{k+1} = x_k - G_k^{-1} g_k. \tag{30}$$

A *Newton direction* is

$$s_k = x_{k+1} - x_k = -G_k^{-1} g_k. \tag{31}$$

We have supposed that G_k is positive definite. So, the Newton direction is a descent direction. This we can conclude from

$$g_k^T s_k = -g_k^T G_k^{-1} g_k < 0.$$

Now, we give the algorithm of the Newton method.

Algorithm 1.2.7. *(Newton method).*
Assumptions: $\epsilon > 0$, $x_0 \in \mathbb{R}^n$. Let $k = 0$.
Step 1. If $\|g_k\| \leq \epsilon$, then STOP.
Step 2. Solve $G_k s = -g_k$ for s_k.
Step 3. Set $x_{k+1} = x_k + s_k$.
Step 4. $k := k + 1$, return to Step 1.
The next theorem shows the local convergence and the quadratic convergence rate of Newton method.

Theorem 1.2.3. [27] *(Convergence theorem of Newton method) Let $f \in C^2$ and x_k be close enough to the solution x^* of the minimization problem with $g(x^*) = 0$. If the Hessian $G(x^*)$ is positively definite and $G(x)$ satisfies Lipschitz condition*

$$|G_{ij}(x) - G_{ij}(y)| \leq \beta \|x - y\|, \text{for some } \beta, \text{for all } i, j,$$

where $G_{ij}(x)$ is the (i,j) element of $G(x)$ and then for all k, Newton direction (31) is well-defined; the generated sequence $\{x_k\}$ converges to x^* with a quadratic rate.

But, in spite of this quadratic rate, the Newton method is a local method: when the starting point is far away from the solution, there is a possibility that G_k is not positive definite, as well as Newton direction is not a descent direction.

So, to guarantee the global convergence, we can use Newton method with line search. We can remind to the fact that only when the step size sequence $\{t_k\}$ tends to 1, Newton method is convergent with the quadratic rate.

Newton iteration with line search is as follows:

$$d_k = -G_k^{-1} g_k, \tag{32}$$

$$x_{k+1} = x_k + t_k d_k. \tag{33}$$

Now, we give the algorithm.

Algorithm 1.2.8. *(Newton method with line search).*
Assumptions: $\epsilon > 0$, $x_0 \in \mathbb{R}^n$. Let $k = 0$.
Step 1. If $\|g_k\| \leq \epsilon$, then STOP.
Step 2. Solve $G_k d = -g_k$ for d_k.
Step 3. Line search step: find t_k such that

$$f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + t d_k),$$

or find t_k such that (inexact) Wolfe line search rules hold.
Step 4. Set $x_{k+1} = x_k + t_k d_k$ and $k = k + 1$, and go to Step 1.
The next theorems claim that Algorithm 1.2.8 with the exact line search, as well as Algorithm 1.2.8 with the inexact line search, are globally convergent.

Theorem 1.2.4. [27] *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on open convex set $D \subset \mathbb{R}^n$. Assume that for any $x_0 \in D$ there exists a constant $m > 0$, such that $f(x)$ satisfies*

$$u^T \nabla^2 f(x) u \geq m \|u\|^2, \text{for all } u \in \mathbb{R}^n, x \in L(x_0), \tag{34}$$

where $L(x_0) = \{x | f(x) \leq f(x_0)\}$ is the corresponding level set. Then, the sequence $\{x_k\}$, generated by Algorithm 1.2.8, with the exact line search, satisfies:

1. When $\{x_k\}$ is a finite sequence, $g_k = 0$ for some k.

2. When $\{x_k\}$ is an infinite sequence, $\{x_k\}$ converges to the unique minimizer x^* of f.

Note that the next relation holds from the standard Wolfe line search:

$$f(x_k) - f(x_k + t_k d_k) \geq \overline{\eta} \|g_k\|^2 \cos^2 \angle(d_k, -g_k), \tag{35}$$

where the constant $\overline{\eta}$ does not depend on k.

Theorem 1.2.5. [27] *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable on open convex set $D \subset \mathbb{R}^n$. Assume that for any $x_0 \in D$ there exists a constant $m > 0$, such that $f(x)$ satisfies the relation (34) on the level set $L(x_0)$. If the line search satisfies the relation (35), then the sequence $\{x_k\}$, generated by Algorithm 1.2.8, with the inexact Wolfe line search, satisfies*

$$\lim_{k \to \infty} \|g_k\| = 0$$

and $\{x_k\}$ converges to the unique minimizer of $f(x)$.

2.4 Modified Newton method

The main problem in Newton method could be the fact that the Hessian G_k may be not positive definite. In that case, we are not sure that the objective function f has its minimizers; furthermore, when G_k is indefinite, the objective function f is unbounded.

So, many modified schemes are made. Now, we describe the next two methods shortly.

In [58], Goldstein and Price use the steepest descent method when G_k is not positive definite. Denoting the angle between d_k and $-g_k$ by θ, as well as having in view the angle rule, $\theta \leq \frac{\pi}{2} - \mu$, where $\mu > 0$, they determine the direction d_k as

$$d_k = \begin{cases} -G_k^{-1} g_k, & \text{if } \cos\theta \geq \eta, \\ -g_k, & \text{otherwise,} \end{cases}$$

where $\eta > 0$ is a given constant.

In [59], the authors present another modified Newton method. When G_k is not positive definite, Hessian G_k is changed into $G_k + \nu_k I$, where $\nu_k > 0$ is chosen in such a way that $G_k + \nu_k I$ is positive definite and well-conditioned. Otherwise, when G_k is positive definite, $\nu_k = 0$.

To consider the other modified Newton methods, such as finite difference Newton method, negative curvature direction method, Gill-Murray stable Newton method, etc., one can see [27], for example.

2.5 Inexact Newton method

By the other side, because of the high cost of the exact Newton method, especially when the dimension n is large, the inexact Newton method might be a good solution. This type of method means that we only approximately solve the Newton equation.

Consider solving the nonlinear equations:

$$F(x) = 0, \tag{36}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is assumed to have the next properties:
$\mathbf{A_1}$ There exists x^* such that $F(x^*) = 0$.
$\mathbf{A_2}$ F is continuously differentiable in the neighborhood of x^*.
$\mathbf{A_3}$ $F'(x^*)$ is nonsingular.
Remind that the basic Newton step is obtained by solving

$$F'(x_k)s_k = -F(x_k)$$

and setting

$$x_{k+1} = x_k + s_k.$$

The inexact Newton method means that we solve

$$F'(x_k)s_k = -F(x_k) + r_k, \tag{37}$$

where

$$\|r_k\| \leq \eta_k \|F(x_k)\|. \tag{38}$$

Set

$$x_{k+1} = x_k + s_k. \tag{39}$$

Here, r_k denotes the residual, and the sequence $\{\eta_k\}$, where $0 < \eta_k < 1$, is the sequence which controls the inexactness.

Now, we give two theorems; the first of them claims the linear convergence, and the second claims the superlinear convergence of the inexact Newton method.

Theorem 1.2.6. [27] *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ satisfy the assumptions A_1–A_3. Let the sequence $\{\eta_k\}$ satisfies $0 \leq \eta_k \leq \eta < t < 1$. Then, for some $\epsilon > 0$, if the starting point is sufficiently near x^*, the sequence $\{x_k\}$ generated by inexact Newton's method (37)–(39) converges to x^*, and the convergence rate is linear, i.e.*

$$\|x_{k+1} - x^*\|_* \leq t\|x_k - x^*\|_*,$$

where $\|y\|_* = \|F'(x^*)y\|$.

Theorem 1.2.7. [27] *Let all assumptions of Theorem 1.2.6 hold. Assume that the sequence $\{x_k\}$, generated by the inexact Newton method, converges to x^*. Then*

$$\|r_k\| = o(\|F(x_k)\|), k \to \infty,$$

if and only if $\{x_k\}$ converges to x^* superlinearly.
The relation

$$x_{k+1} = x_k - \frac{f'(x_k)}{f'(x_k) - f'(x_{k-1})} \cdot (x_k - x_{k-1}), \tag{40}$$

presents the secant method.

In [60], a modification of the classical secant method for solving nonlinear, univariate, and unconstrained optimization problems based on the development of the cubic approximation is presented. The iteration formula including an approximation of the third derivative of $f(x)$ by using the Taylor series expansion is derived. The basic assumption on the objective function $f(x)$ is that $f(x)$ is a real-valued function of a single, real variable x and that $f(x)$ has a minimum at x^*. Furthermore, in this chapter it is noted that the secant method is the simplification of Newton method. But, the order of the secant method is lower than one of the Newton methods; it is Q-superlinearly convergent, and its order is $p = \frac{\sqrt{5}+1}{2} \approx 1,618$.

This modified secant method is constructed in [60], having in view, as it is emphasized, that it is possible to construct a cubic function which agrees with $f(x)$ up to the third derivatives. The third derivative of the objective function f is approximated as

$$f'''(x) = \frac{3\left\{\frac{2\left[f'(x_k) - \frac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}\right]}{x_k - x_{k-1}} - f''(x_k)\right\}}{x_{k-1} - x_k}.$$

In [61], the authors propose an inexact Newton-like conditional gradient method for solving constrained systems of nonlinear equations. The local convergence of the new method as well as results on its rate is established by using a general majorant condition.

2.6 Quasi-Newton method

Consider the Newton method.

For various practical problems, the computation of Hessian may be very expensive, or difficult, or Hessian can be unavailable analytically. So, the class of so-called quasi-Newton methods is formed, such that it uses only the objective function values and the gradients of the objective function and it is close to Newton method. Quasi-Newton method is such a class of methods which does not compute Hessian, but it generates a sequence of Hessian approximations and maintains a fast rate of convergence.

So, we would like to construct Hessian approximation B_k in quasi-Newton method. Naturally, it is desirable that the sequence $\{B_k\}$ possesses positive definiteness, as well as its direction $d_k = -B_k^{-1}g_k$ should be a descent one.

Now, let $f : \mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable function on an open set $D \subset \mathbb{R}^n$. Consider the quadratic approximation of f at x_{k+1}:

$$f(x) \approx f(x_{k+1}) + g_{k+1}^T(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T G_{k+1}(x - x_{k+1}).$$

Finding the derivatives, we get

$$g(x) \approx g_{k+1} + G_{k+1}(x - x_{k+1}).$$

Setting $x = x_k$ and using the standard notation: $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$, from the last relation, we get

$$G_{k+1}^{-1} y_k \approx s_k. \tag{41}$$

Relation (41) transforms into the next one if f is the quadratic function:

$$G_{k+1}^{-1} y_k = s_k. \tag{42}$$

Let H_k be the approximation of the inverse of Hessian. Then, we want H_k to satisfy the relation (42). In this way, we come to the quasi-Newton condition or quasi-Newton equation:

$$H_{k+1} y_k = s_k. \tag{43}$$

Let $B_{k+1} = H_{k+1}^{-1}$ be the approximation of Hessian G_{k+1}. Then

$$B_{k+1} s_k = y_k \tag{44}$$

is also the quasi-Newton equation.
If

$$s_k^T y_k > 0, \tag{45}$$

then the matrix B_{k+1} is positive definite. The condition (45) is known as the curvature condition.

Algorithm 1.2.9. *(A general quasi-Newton method).*
Assumptions: $0 \leq \epsilon < 1$, $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$. Let $k = 0$.
Step 1. If $\|g_k\| \leq \epsilon$, then STOP.
Step 2. Compute $d_k = -H_k g_k$.
Step 3. Find t_k by line search and set $x_{k+1} = x_k + t_k d_k$.
Step 4. Update H_k into H_{k+1} such that quasi-Newton equation (43) holds.
Step 5. Set $k = k + 1$ and go to Step 1.
In Algorithm 1.2.9, usually we take $H_0 = I$, where I is an identity matrix.
Sometimes, instead of H_k, we use B_k in Algorithm 1.2.9.
Then, *Step* 2 becomes
Step 2*. Solve

$$B_k d = -g_k, \text{ for } d_k.$$

By the other side, *Step* 4 becomes
Step 4*. Update B_k into B_{k+1} in such a way that quasi-Newton equation (44) holds.

2.7 Symmetric rank-one (*SR*1) update

Let H_k be the inverse Hessian approximation of the kth iteration. We are trying to update H_k into H_{k+1}, i.e.

$$H_{k+1} = H_k + E_k,$$

where E_k is a matrix with a lower rank. If it is about a rank-one update, we get

$$H_{k+1} = H_k + uv^T, \tag{46}$$

where $u, v \in \mathbb{R}^n$. Using quasi-Newton equation (43), we can get

$$H_{k+1} y_k = \left(H_k + uv^T\right) y_k = s_k,$$

wherefrom

$$\left(v^T y_k\right)u = s_k - H_k y_k. \tag{47}$$

Further, from (46) and (47), we have

$$H_{k+1} = H_k + \frac{1}{v^T y_k}\left(s_k - H_k y_k\right)v^T.$$

Having in view that the inverse Hessian approximation H_k has to be the symmetric one, we use $v = s_k - H_k y_k$, so we get the symmetric rank-one update (i.e., *SR*1 update):

$$H_{k+1} = H_k + \frac{\left(s_k - H_k y_k\right)\left(s_k - H_k y_k\right)^T}{\left(s_k - H_k y_k\right)^T y_k}. \tag{48}$$

Theorem 1.2.8. [27] *(Property theorem of SR1 update) Let s_0, s_1, and s_{n-1} be linearly independent. Then, for quadratic function with a positive definite Hessian, SR1 method terminates at $n+1$ steps, i.e., $H_n = G^{-1}$.*

More information about SR1 update can be found.

2.8 Davidon-Fletcher-Powell (*DFP*) update

There exists another type of update, which is a rank-two update. In fact, we get H_{k+1} using two symmetric, rank-one matrices:

$$H_{k+1} = H_k + auu^T + bvv^T, \tag{49}$$

where $u, v \in \mathbb{R}^n$ and a, b are scalars which have to be determined.

Using quasi-Newton equation (43), we can get

$$H_k y_k + auu^T y_k + bvv^T y_k = s_k. \tag{50}$$

The values of u, v are not determined in a unique way, but the good choice is

$$u = s_k, v = H_k y_k.$$

Now, from (50), we get:

$$a = \frac{1}{s_k^T y_k}, b = -\frac{1}{y_k^T H_k y_k}.$$

Hence, we get the formula

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}, \tag{51}$$

which is *DFP* update.

Theorem 1.2.9. [27] *(Positive definiteness of DFP update) DFP update (51) retains positive definiteness if and only if $s_k^T y_k > 0$.*

Theorem 1.2.10. [27] *(Quadratic termination theorem of DFP method) Let $f(x)$ be a quadratic function with positive definite Hessian G. Then, if the exact line search is used, the sequence $\{s_j\}$, generated from DFP method, satisfies, for $i = 0, 1, ..., m$, where $m \leq n-1$:*

1. $H_{i+1}y_j = s_j, j = 0, 1, ..., i$ (*hereditary property*).

2. $s_i^T G s_j = 0, j = 0, 1, ..., i-1$ (*conjugate direction property*).

3. *The method terminates at* $m+1 \leq n$ *steps. If* $m = n-1$, *then* $H_n = G^{-1}$.

2.9 Broyden-Fletcher-Goldfarb-Shanno (*BFGS*) update

BFGS update is given by the formula

$$B_{k+1}^{BFGS} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}. \tag{52}$$

The *BFGS* update is also said to be a complement to *DFP* update.

In [62], an adaptive scaled *BFGS* method for unconstrained optimization is presented. In this paper, the author emphasizes that the *BFGS* method is one of the most efficient quasi-Newton methods for solving small-size and medium-size unconstrained optimization problems. The third term in the standard *BFGS* update formula is scaled in order to reduce the large eigenvalues of the approximation to the Hessian of the minimizing function. In fact, in [62], the general scaling *BFGS* updating formula is considered:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}, \tag{53}$$

where γ_k is a positive parameter. Obviously, using $\gamma_k = 1$ for all $k = 0, 1, ...$, we get the standard *BFGS* formula. By the way, there exist several procedures created to select the scaling parameter γ_k, for example, see [62–69]. The approach for determining the scaling parameters of the terms of the *BFGS* update in [62] is to minimize the Byrd and Nocedal measure function.

Namely, in [70], the next function was introduced:

$$\varphi(A) = tr(A) - \ln(\det(A)), \tag{54}$$

which is defined on positive definite matrices.

This function is a measure of matrices involving all the eigenvalues of A, not only the smallest one and the largest one, as it is traditionally used in the analysis of the quasi-Newton method based on the condition number of matrices.

Observe that function φ works simultaneously with the trace and the determinant, thus simplifying the analysis of the quasi-Newton methods. Fletcher [71] proves that this function is strictly convex on the set of symmetric and positive definite matrices, and it is minimized by $A = I$. Besides, this function becomes unbounded when A becomes singular or infinite, and therefore it works as a barrier function that keeps A positive definite. It is worth saying that the *BFGS* update tends to generate updates with large eigenvalues.

Further, in [62], a double-parameter scaling *BFGS* update is considered, in which the first two terms on the right-hand side of the *BFGS* update (52) are scaled with a positive parameter, while the third one is scaled with another positive parameter:

$$B_{k+1} = \delta_k \left[B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \right] + \gamma_k \frac{y_k y_k^T}{y_k^T s_k}, \tag{55}$$

where δ_k and γ_k are the two positive parameters that have to be determined.

In [62], the next proposition is proved.

Proposition 1.2.1. *If the step size t_k is determined by the standard Wolfe line search (12) and (13), B_k is positive definite and $\gamma_k > 0$, and then B_{k+1}, given by (55), is also positive definite.*

From (55), it can be seen that $\varphi(B_{k+1})$ depends on the scaling parameters δ_k and γ_k. In [62], these scaling parameters are determined as solution of the minimizing problem:

$$\min_{\delta_k > 0,\, \gamma_k > 0} \varphi(B_{k+1}). \tag{56}$$

Further, the next values of the scaling parameters δ_k and γ_k are reached:

$$\delta_k = \frac{n-1}{tr(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k}} \tag{57}$$

$$\gamma_k = \frac{y_k^T s_k}{\|y_k\|^2}. \tag{58}$$

Consider the relation

$$x_{k+1} = x_k + t_k d_k, \tag{59}$$

where d_k is the *BFGS* search direction obtained as solution of the linear algebraic system

$$B_k d_k = -g_k,$$

where the matrix B_k is the *BFGS* approximation to the Hessian $\nabla^2 f(x_k)$, being updated by the classical formula (52).

The next theorems are also given in [62].

Theorem 1.2.11. *If the step size in (59) is determined by the Wolfe search conditions (12)–(13), then the scaling parameters given by (57) and (58) are the unique global solutions of the problem (56).*

Theorem 1.2.12. *Let δ_k be computed by (57). Then, for any $k = 0, 1, \ldots$, δ_k is positive and close to* 1.

Next, in [72], using chain rule, a modified secant equation is given, to get a more accurate approximation of the second curvature of the objective function. Then, based on this modified secant equation, a new *BFGS* method is presented. The proposed method makes use of both gradient and function values, and it utilizes information from two most recent steps, while the usual secant relation uses only the latest step information. Under appropriate conditions, it is shown that the proposed method is globally convergent without convexity assumption on the objective function.

Some interesting applications of Newton, modified Newton, inexact Newton, and quasi-Newton methods can be found, for example, in [73–83], etc.

A very interesting paper is [84].

An interesting application of *BFGS* method can be found in [85].

3. Conclusion

Today, the modifications of the line search techniques are very actual and all in the aim to create new, better optimization methods.

Further, following recent trends in unconstrained optimization, we can notice that almost all optimization methods, which are considered in this chapter, are still actual.

They are applied in the other areas of Mathematics, as well as in practice. Also, different modifications of these methods are made, in the aim to improve them.

Let us emphasize that *BFGS* update is very popular now.

Author details

Snezana S. Djordjevic
Faculty of Technology, University of Nis, Leskovac, Serbia

*Address all correspondence to: snezanadjordjevicle@gmail.com

References

[1] Armijo L. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of Mathematics. 1966;**16**(1):1-3

[2] Goldstein AA. On steepest descent. SIAM Journal on Control and Optimization. 1965;**3**:147-151

[3] Wolfe P. Convergence conditions for ascent methods. SIAM Review. 1969;**11**: 226-235

[4] Wolfe P. Convergence conditions for ascent methods. II: Some corrections. SIAM Review. 1969;**11**:226-235

[5] Nocedal J, Wright SJ. Numerical Optimization. New York, NY, USA: Springer Verlag; 2006

[6] Krejic N, Jerinkic NK. Nonmonotone line search methods with variable sample size. Numerical Algorithms. 2015;**68**(4):711-739

[7] Dai YH. On the nonmonotone line search. Journal of Optimization Theory and Applications. 2002;**112**:315-330

[8] Raydan M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM Journal on Optimization. 1997;**7**: 26-33

[9] Toint PhL. Nonmonotone trust region algorithms for nonlinear optimization subject to convex constraints. Mathematical Programming. 1997;**77**:69. DOI: 10.1007/BF02614518

[10] Toint PL. An assessment of non-monotone line search techniques for unconstrained optimization. SIAM Journal on Scientific Computing. **17**(3): 725-739. 15 pages

[11] Zhang H, Hager W. A nonmonotone line search technique and its application to unconstrained optimization. SIAM Journal on Optimization. 2004;**4**: 1043-1056

[12] Grippo L, Lampariello F, Lucidi S. A nonmonotone line search technique for Newton's method. SIAM Journal on Numerical Analysis. 1986;**23**:707-716

[13] Li DH, Fukushima M. A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optimization Methods and Software. 2000;**13**:181-201

[14] Birgin EG, Krejic N, Martinez JM. Globally convergent inexact Quasi-Newton methods for solving nonlinear systems. Numerical Algorithms. 2003; **32**:249-250

[15] SHI Z-J, Shen J. Convergence of descent method with new line search. Journal of Applied Mathematics and Computing. 2006;**20**(1–2):239-254

[16] Wan et al. New cautious BFGS algorithm based on modified Armijo-type line search. Journal of Inequalities and Applications. 2012;**2012**:241

[17] Yu G, Guan L, Wei Z. Globally convergent Polak-Ribiére-Polyak conjugate gradient methods under a modified Wolfe line search. Applied Mathematics and Computation. 2009; **215**:3082-3090

[18] Huang S, Wan Z, Zhang J. An extended nonmonotone line search technique for large-scale unconstrained optimization. Journal of Computational and Applied Mathematics. 2018;**330**: 586. 19p

[19] Koorapetse MS, Kaelo P. Globally convergent three-term conjugate

gradient projection methods for solving nonlinear monotone equations. Arabian Journal of Mathematics. 2018;**7**(4): 289-301

[20] Yu Z, Pu D. A new nonmonotone line search technique for unconstrained optimization. Journal of Computational and Applied Mathematics. 2008;**219**: 134-144

[21] Yuan G, Wei Z. A modified PRP conjugate gradient algorithm with nonmonotone line search for nonsmooth convex optimization problems. Journal of Applied Mathematics and Computing. 2016;**51**: 397-412

[22] Yuan G, Wei Z, Lu X. Global convergence of the BFGS method and the PRP method for general functions under a modified weak Wolfe-Powell line search. Applied Mathematical Modelling. 2017;**47**: 811-825

[23] Yuan G, Wei Z, Yang Y. The global convergence of the Polak-Ribiére-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions. Journal of Computational and Applied Mathematics. 2018. DOI: 10.1016/j.cam.2018.10.057. In press

[24] Cauchy A. Méthode générale pour la résolution des systéms d'equations simultanées. Comptes Rendus Mathematique Academie des Sciences, Paris. 1847;**25**:46-89

[25] Johari R, Rivaie M, Mamat M. A new scaled steepest descent method for unconstrained optimization with global convergence properties. Journal of Engineering and Applied Sciences. 2018; **13**(Special Issue 6):5442-5445

[26] Wen GK, Mamat M, Mohd IB, Dasril Y. A novel of step size selection procedures for steepest descent method. Applied Mathematical Sciences. 2012; **6**(51):2507-2518

[27] Sun W, Yuan Y-X. Optimization theory and methods: Nonlinear programming. Springer: Optimization and Its Applications. 2006

[28] Abidin ZAZ, Mamat M, Rivaie M, Mohd I. A new steepest descent method. In: Proceedings of the 3rd International Conference on Mathematical Sciences, Vol. 1602, December 17–19; Melville, New York: AIP; 2013. pp. 273-278

[29] Andrei N. Relaxed Gradient Descent and a New Gradient Descent Methods for Unconstrained Optimization. Available from: https://camo.ici.ro/neculai/newgrad.pdf

[30] Knyazev AV, Lashuk I. Steepest descent and conjugate gradient methods with variable preconditioning. SIAM Journal on Matrix Analysis and Applications. 2007;**29**(4):1267-1280

[31] Liu C-S, Chang J-R, Chen Y-W. A modified algorithm of steepest descent method for solving unconstrained nonlinear optimization problems. Journal of Marine Science and Technology. 2015;**23**(1):88-97

[32] Osadcha O, Marszaek Z. Comparison of Steepest Descent Method and Conjugate Gradient Method. Available from: http://ceur-ws.org/Vol-1853/p01.pdf

[33] Andrei N. Scaled conjugate gradient algorithms for unconstrained optimization. Computational Optimization and Applications. 2007;**38**: 401-416

[34] Napitupulu et al. Steepest descent method implementation on unconstrained optimization problem using C++ program. IOP Conference

Series: Materials Science and Engineering. 2018;**332**:012024

[35] Yuan Y. A new stepsize for the steepest descent method. Journal of Computational Mathematics. 2006; **24**(2):149-156

[36] Raydan M, Svaiter B. Relaxed steepest descent and Cauchy-Barzilai-Borwein method. Computational Optimization and Applications. 2002; **21**(2):155-167

[37] Cai Y, Bai Z, Pask JE, Sukumar N. Convergence analysis of a locally accelerated preconditioned steepest descent method for hermitian-definite generalized eigenvalue problems. Journal of Computational Mathematics. 2018;**36**(5):739-760

[38] Egorova I, Michor J, Teschl G. Rarefaction waves for the Toda equation via nonlinear steepest descent. Discrete and Continuous Dynamical Systems. 2018;**38**:2007-2028

[39] Gonzaga CC. On the worst case performance of the steepest descent algorithm for quadratic functions. Mathematical Programming, Series A. 2016;**160**:307-320

[40] Hosokawa S, Pusztai L, Matsushita T. Algorithm for atomic resolution holography using modified *L*1-regularized linear regression and steepest descent method. Physica Status Solidi B: Basic Solid State Physics. 2018; **255**:11

[41] Liu X, Reynolds AC. A multiobjective steepest descent method with applications to optimal well control. Computational Geosciences. 2016;**20**:355-374

[42] Svaiter BF. Hölder continuity of the steepest descent direction for multiobjective optimization. 2018. arXiv:1802.01402v1 [math.OC]

[43] Torres P, van Wingerden J-W. Identification of 2D interconnected systems: An efficient steepest-descent approach. IFAC Papers OnLine. 2018; **51**(15):78-83

[44] Barzilai J, Borwein J. Two-point step size gradient methods. IMA Journal of Numerical Analysis. 1988;**8**(1):141-148

[45] Grippo L, Sciandrone M. Nonmonotone globalization techniques for the Barzilai-Borwein gradient method. Computational Optimization and Applications. 2002;**23**:143-169

[46] Dai YH, Hager WW, Schittkowski K, Zhang H. The cyclic Barzilai-Borwein method for unconstrained optimization. IMA Journal of Numerical Analysis. 2006;**26**:604-627

[47] Raydan M. On the Barzilai and Borwein choice of steplength for the gradient method. IMA Journal of Numerical Analysis. 1993;**13**(3):321-326

[48] Dai Y, Liao L. R-linear convergence of the Barzilai and Borwein gradient method. IMA Journal of Numerical Analysis. 2002;**22**(1):1-10

[49] Friedlander A, Martinez JM, Molina B, Raydan M. Gradient method with retards and generalizations. SIAM Journal on Numerical Analysis. 1999;**36**: 275-289

[50] Narushima Y, Wakamatsu T, Yabe H. Extended Barzilai-Borwein method for unconstrained minimization problems. Pacific Journal of Optimization. 2008;**6**(3):591-614

[51] Dai YH, Yuan Y. Analysis of monotone gradient methods. Journal of Industrial and Management Optimization. 2005;**1**:181-192

[52] Fletcher R. On the Barzilai-Borwein method, Optimization and Control with Applications. Springer Series in Applied Optimization 96. New York: Springer-Verlag; 2005. pp. 235-256

[53] Liu ZX, Liu HW. An efficient gradient method with approximate optimal stepsize for large-scale unconstrained optimization. Numerical Algorithms. 2018;**78**(1):21-39

[54] La Cruz W. A spectral algorithm for large-scale systems of nonlinear monotone equations. Numerical Algorithms. 2017;**76**:1109-1130

[55] Feng X, Hormuth DA II, Yankeelov TE. An adjoint-based method for a linear mechanically-coupled tumor model: Application to estimate the spatial variation of murine glioma growth based on diffusion weighted magnetic resonance imaging. Computational Mechanics. 2018. DOI: 10.1007/s00466-018-1589-2

[56] Krzysztof S, Drozda Stochastic P. Gradient descent with Barzilai-Borwein update step for *SVM*. Information Sciences. 2015;**316**:218-233

[57] Li M, Liu H, Liu Z. A new subspace minimization conjugate gradient method with nonmonotone line search for unconstrained optimization. Numerical Algorithms. 2018;**79**:195-219

[58] Goldstein AA, Price JF. An effective algorithm for minimization. Numerische Mathematik. 1967;**10**:184-189

[59] Goldfeld SM, Quandt RE, Trotter HF. Maximisation by quadratic hill-climbing. Econometrica. 1966;**34**: 541-551

[60] Kahya E, Chen J. A modified Secant method for unconstrained optimization. Applied Mathematics and Computation. 2007;**186**:1000-1004

[61] Gonçalves MLN, Oliveira FR. An inexact Newton-like conditional gradient method for constrained nonlinear systems. Applied Numerical Mathematics. 2018;**132**:22-34

[62] Andrei N. An adaptive scaled BFGS method for unconstrained optimization. Numerical Algorithms. 2018;**77**(2): 413-432

[63] Andrei N. A double parameter scaled BFGS method for unconstrained optimization. Journal of Computational and Applied Mathematics. 2018;**332**:26-44

[64] Biggs MC. Minimization algorithms making use of non-quadratic properties of the objective function. Journal of the Institute of Mathematics and its Applications. 1971;**8**:315-327

[65] Biggs MC. A note on minimization algorithms making use of non-quadratic properties of the objective function. Journal of the Institute of Mathematics and its Applications. 1973; **12**:337-338

[66] Liao A. Modifying *BFGS* method. Operations Research Letters. 1997;**20**: 171-177

[67] Nocedal J, Yuan YX. Analysis of self-scaling quasi-Newton method. Mathematical Programming. 1993;**61**: 19-37

[68] Oren SS, Luenberger DG. Self-scaling variable metric (SSVM) algorithms, Part I: Criteria and sufficient conditions for scaling a class of algorithms. Management Science. 1974; **20**:845-862

[69] Yuan YX. A modified BFGS algorithm for unconstrained optimization. IMA Journal of Numerical Analysis. 1991;**11**:325-332

[70] Byrd R, Nocedal J. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. SIAM Journal on Numerical Analysis. 1989;**26**:727-739

[71] Fletcher R. An overview of unconstrained optimization. In: Spedicato E, editor. Algorithms for Continuous Optimization: The State of the Art. Boston: Kluwer Academic Publishers; 1994. pp. 109-143

[72] Dehghani R, Bidabadi N, Hosseini MM. A new modified BFGS method for unconstrained optimization problems. Computational and Applied Mathematics. 2018;**37**:5113-5125

[73] Andrei N. A diagonal quasi-Newton updating method for unconstrained optimization. Numerical Algorithms. 2018:16. DOI: 10.1007/s11075-018-0562-7. In press

[74] Bajović D, Jakovetić D, Krejić N, Krklec Jerinkić N. Newton-like method with diagonal correction for distributed optimization. SIAM Journal on Optimization. 2017;**27**(2):1171-1203

[75] Carraro T, Dörsam S, Frei S, Schwarz D. An adaptive newton algorithm for optimal control problems with application to optimal electrode design. Journal of Optimization Theory and Applications. 2018;**177**:498-534

[76] Djordjević SS. Two modifications of the method of the multiplicative parameters in descent gradient methods. Applied Mathematics and Computation. 2012;**218**(17):8672-8683

[77] Ferreira OP, Silva GN. Inexact Newton method for non-linear functions with values in a cone. Applicable Analysis. 2018. https://www.tandfonline.com/doi/abs/10.1080/00036811.2018.1430779

[78] Grapsa TN. A modified Newton direction for unconstrained optimization. A Journal of Mathematical Programming and Operations Research. 2014;**63**(7):983-1004

[79] Li Y-M, Guo X-P. On the accelerated modified Newton-HSS method for systems of nonlinear equations. Numerical Algorithms. 2018;**79**:1049-1073

[80] Matebese B, Withey D, Banda MK. Modified Newton's method in the leapfrog method for mobile robot path planning. In: Dash S, Naidu P, Bayindir R, Das S, editors. Artificial Intelligence and Evolutionary Computations in Engineering Systems. Advances in Intelligent Systems and Computing. Vol. 668. Singapore: Springer; 2018. pp. 71-78

[81] Mezzadri F, Galligani E. An inexact Newton method for solving complementarity problems in hydrodynamic lubrication. Calcolo. 2018;**55**:1

[82] Sharma JR, Argyros IK, Kumar D. Newton-like methods with increasing order of convergence and their convergence analysis in Banach space. SeMA. 2018;**75**:545-561

[83] Stanimirović P, Miladinović M, Djordjević S. Multiplicative parameters in gradient descent methods. Univerzitet u Nišu. 2009;**23**(3):23-36

[84] Petrović MJ, Stanimirović PS, Kontrec N, Mladenov J. Hybrid modification of accelerated double direction method. Mathematical Problems in Engineering. 2018;**2018**:1-8

[85] Stanimirovic PS, Ivanov B, Djordjevic S, Brajevic I. New hybrid conjugate gradient and Broyden-Fletcher-Goldfarb-Shanno conjugate gradient methods. Journal of Optimization Theory and Applications. 2018;**178**(3):860-884