DERIVATIVE FREE ALGORITHMS FOR LARGE SCALE NON-SMOOTH OPTIMIZATION AND THEIR APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ HAKAN TOR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MATHEMATICS

FEBRUARY 2013

Approval of the thesis:

DERIVATIVE FREE ALGORITHMS FOR LARGE SCALE NON-SMOOTH OPTIMIZATION AND THEIR APPLICATIONS

submitted by **ALİ HAKAN TOR** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mathematics Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ____________

Prof. Dr. Mustafa Korkmaz
Head of Department, **Mathematics** ____________

Prof. Dr. Bülent Karasözen
Supervisor, **Mathematics Department, METU** ____________

Assoc. Prof. Dr. Adil Bagirov
Co-supervisor, **Graduate School of ITMS, University of Ballarat** ____________

Examining Committee Members:

Prof. Dr. Gerhard Wilhelm Weber
Institute of Applied Mathematics, METU ____________

Prof. Dr. Bülent Karasözen
Mathematics Department, METU ____________

Prof.Dr. Refail Kasımbeyli
Industrial Engineering Department, Anadolu University ____________

Assoc. Prof. Dr. Songül Kaya Merdan
Mathematics Department, METU ____________

Assoc.Prof.Dr. Ömür Uğur
Institute of Applied Mathematics, METU ____________

Date: ____________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ALİ HAKAN TOR

Signature :

ABSTRACT

DERIVATIVE FREE ALGORITHMS FOR LARGE SCALE NON-SMOOTH OPTIMIZATION AND THEIR APPLICATIONS

Tor, Ali Hakan

Ph.D., Department of Mathematics

Supervisor : Prof. Dr. Bülent Karasözen

Co-Supervisor : Assoc. Prof. Dr. Adil Bagirov

February 2013, 88 pages

In this thesis, various numerical methods are developed to solve nonsmooth and in particular, nonconvex optimization problems. More specifically, three numerical algorithms are developed for solving nonsmooth convex optimization problems and one algorithm is proposed to solve nonsmooth nonconvex optimization problems.

In general, main differences between algorithms of smooth optimization are in the calculation of search directions, line searches for finding step-sizes and stopping criteria. However, in nonsmooth optimization there is one additional difference between algorithms. These algorithms may use different generalizations of the gradient. In order to develop algorithms for solving nonsmooth convex optimization problems we use the concept of codifferential. Although there exists the codifferential calculus, the calculation of the whole codifferential is not an easy task. Therefore, in the first numerical method, only a few elements of the codifferential are used to calculate search directions. In order to reduce the number of codifferential evaluations, in the second method elements of the codifferential calculated in previous iterations are used to calculate search directions.

In both the first and second methods the problem of calculation of search directions is reduced to the solution of a certain quadratic programming problem. The size of this problem can increase significantly as the number of variables increases. In order to avoid this problem in the third method, called the aggregate codifferential method, the number of elements of the codifferential used to find search directions is fixed. Such an approach allows one to significantly reduce the complexity of codifferential methods and to make them applicable for solving large scale problems of nonsmooth optimization.

These methods are applied to some well-known nonsmooth optimization test problems, such as, minmax and general type nonsmooth optimization problems. The obtained numerical results are visualized using performance profiles. In addition, the validation of these methods is made by comparing them with the subgradient and bundle methods using results of numerical experiments. The convergence of methods is analyzed. Finally, the first method is extended to minimize nonsmooth convex functions subject to linear inequalities using slack variables.

The notion of quasisecant is used to design an algorithm for solving nonsmooth nonconvex unconstrained optimization problems. In this method, to find descent direction the subgradient algorithm is applied for the solution of a set of linear inequalities. The convergence of the proposed method is analyzed, and the numerical experiments are carried out using general type nonsmooth optimization test problems. To validate this method, the results are compared with those by the subgradient method.

Keywords: Nonsmooth optimization, convex optimization, nonconvex optimization, codifferential, subdifferential.

ÖZ

TÜREVİ KULLANMAYAN OPTİMİZASYON YÖNTEMLERİNİN, ÇOK BOYUTLU TÜREVİ OLMAYAN OPTİMİZASYON PROBLEMLERİNE UYGULANMASI

Tor, Ali Hakan
Doktora, Matematik Bölümü
Tez Yöneticisi : Prof. Dr. Bülent Karasözen
Ortak Tez Yöneticisi : Doç. Dr. Adil Bagirov

Şubat 2013, 88 sayfa

Bu tezin amacı türevi olmayan optimizasyon problemlerini çözmek için yöntem geliştirmektir ve türevi olmayan optimizasyon problemleri iki bölümde incelenmiştir; dışbükey ve dışbükey olmayan optimizasyon problemleri. Bu tezde bu iki tip problem için yöntemler geliştirilmiştir.

İlk olarak türevi olmayan kısıtsız dışbükey optimizasyon problemler için kodiferansiyel kavramı kullanarak üç farklı yöntem geliştirilmiştir. Bilindiği gibi, aynı tip optimizasyon problemlerini çözmek için geliştirilen algoritmalar, türev yerine kullanılan kavram, durma kriterleri ve azalma yönü hesaplarına göre farklılaşmaktadırlar. Bu tezde geliştirilen bu üç metotta ise türev yerine kodiferansiyel kullanılmıştır. Kodiferansiyelin yapısı gereği durma kriterleri bu üç metotta da aynıdır. Diğer taraftan, azalma yönü hesaplanmasına baktığımızda metotlar farklılıklar göstermektedir. Bu farklılıklar şu şekilde sıralanmaktadır. Yöntemlerden birincisinde azalma yönü kodiferansiyelin sadece bazı elemanlarını kullanarak hesaplanmaktadır. İkincisinde ise, fonksiyon ve gradient hesaplamalarının sayısını azaltmak için bir önceki basamakta elde edilen kodiferansiyel değerleri kullanılmıştır. Son metotta ise azalma yönü her iterasyonda sabit ve belli sayıda kodiferansiyelleri kullanarak hesaplanmaktadır. Bunun yanında, geliştirilen yöntemlerin yakınsaklık analizleri yapılmıştır. Bu yöntemler literatürde bilinen önemli test problemlerine uygulanmış ve elde edilen sayısal sonuçlar performans grafikleri ile gösterilmiştir. Bu grafikler, bilinen alt-gradient ve demet yöntemleriyle de elde edilen performans grafikleriyle karşılaştırılmıştır ve geliştirmiş olduğumuz metotların daha iyi sonuç verdiği gözlemlenmiştir. Bunların yanında, yukarıda bahsi geçen ilk metodun yapay değişkenler kullanarak uyarlanan yeni hali, doğrusal kısıtlı dışbükey optimizasyon problemlerine uygulanmıştır. Uygulama olarak üç test problemi alınmış ve sayısal sonuçlar tablolar kullanılarak gösterilmiştir.

Son olarak, dışbükey olmayan optimizasyon problemleri için yöntem geliştirilmiştir. “Quasisecant” kavramı kullanılarak geliştirilen bu yöntemde, azalma yönü hesabı için bir alt-gradient yöntemi kullanılarak doğrusal eşitsizlik sistemi çözülmüştür. Geliştirilen bu yöntemin yakınsaklığı incelenmiş, bilinen bazı önemli test problemleri kullanılarak sayısal hesaplamalar yapılmış ve bu sonuçlar bir alt-gradient yöntemiyle kıyaslanarak bir tabloda sunulmuştur.

Anahtar Kelimeler: Türevi olmayan optimizasyon, dışbükey optimizasyon, dışbükey olmayan optimizasyon, kodiferansiyel, alt-gradient.

To my daughter Tılsım
my wife Dürdane
my parents İnci and Şükrü
my sister Yeşim

ACKNOWLEDGMENTS

I would like to thank many people who have helped me through the completion of this dissertation. A Ph.D. thesis would not have been possible without supervisor(s) so that it is the first pleasure to thank my supervisors Prof. Dr. Bülent Karasözen and Assoc. Prof. Dr. Adil Bagirov who made this thesis possible. I am heartily thankful to my supervisor, Prof. Dr. Bülent Karasözen, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject. On the other hand, I am also grateful to him for introducing me to my co-supervisor, Assoc. Prof. Dr. Adil Bagirov. I cannot find words to express my gratitude to Assoc. Prof. Bagirov. It is with immense gratitude that I acknowledge guidance, continuous encouragement and support and help of him. Besides, I thank his hospitality for my two visit to Ballarat University. I think that to meet him is a big chance not only for my thesis but also for my academic life. In addition, many thanks to Prof. Dr. Gerhard Wilhelm Weber and Assoc. Prof. Dr. Canan Bozkaya who kindly gave their time to review and edit my thesis. I would also like to thank to extend my appreciation to the committee members, Prof. Dr. Gerhard Wilhelm Weber, Prof.Dr. Refail Kasımbeyli, Assoc. Prof. Dr. Songül Kaya Merdan, and Assoc. Prof. Dr. Ömür Uğur, for their valuable suggestions and reviewing my work.

I also would like to express my sincere thanks to the Prof. Dr. Zafer Nurlu, former head of department of mathematics at METU, and Prof. Dr. Mustafa Korkmaz, current head of department of mathematics at METU for their support to enable all bureaucratic process easily during my PhD adventure.

The financial support of The Scientific and Technical Research Council of Turkey (TÜBİTAK) is also acknowledged.

I am grateful to all my friends, namely Arzu, Canan, Hamdullah, Nagehan, Nuray, Nil, Onur and Zeynep, who have given me help and encouragement with everything that was involved in doing this research.

Special thanks to my mother, İnci, my father, Şükrü, and my sister Yeşim, for their love, continuing moral and emotional support, and their help which makes PhD process easy.

Last, but not the least, my heartfelt appreciation goes to my wife, who is the meaning of existences of me and our lovely child Tılsım, for her encouragement, loving support, understanding and patience.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF NOTATIONS

The following notation will be used in this thesis.

$\langle u, v\rangle = \sum_{i=1}^{n} u_i v_i$	:	inner product in $\mathbb{R}^n$
$\|\cdot\|$	:	associated Euclidean norm
$\mathrm{co}\,\cdot$	:	convex hull of a set
$\mathrm{cl}\,\cdot$	:	closure of a set topologically
$\mathbb{R}^n$	:	n-dimensional Euclidean space
$S_1 = \{x \in \mathbb{R}^n : \|x\| = 1\}$	:	the unit sphere
$B_\varepsilon(x) = \{y \in \mathbb{R}^n : \|y - x\| < \varepsilon\}$	:	open ball centered at x with the radius $\varepsilon > 0$
$\bar{B}_\varepsilon(x) = \mathrm{cl}\, B_\varepsilon(x)$	:	closed ball centered at x with the radius $\varepsilon > 0$
$\mathrm{Rank}\,\cdot$	:	rank of a matrix
A^T	:	transpose of a matrix A
argmin$\cdot$	:	the argument of the minumum, that is to say, the set of points of the given argument for which the given function attains its maximum value
argmax$\cdot$	:	the argument of the maximum, that is to say, the set of points of the given argument for which the given function attains its maximum value

CHAPTER 1

INTRODUCTION

Optimization theory deals with the finding of local or global minimizers of a function on a given set. The function, whose minimum is being sought, is called the objective function, and the function(s), which describe the set where the local or global minimizer is being sought, are called constraints. There is an advanced theory when the objective and constraint functions are continuously differentiable. Powerful methods have been advanced to solve smooth optimization problems. If at least one of those functions is not continuously differentiable then the optimization is said to be nonsmooth. Algorithmic developments in nonsmooth optimization are far being mature. Unlike smooth optimization the finding a descent direction and evaluation of optimality conditions are not easy task. Thus, researchers are interested in designing efficient numerical methods for nonsmooth problems which have been motivated by practical applications from different areas. To illustrate, in Economics, tax models consist of several different structures which are not continuously differentiable at their intersections. In steel industry, the material changes the phase discontinuously because of the nature of the material. In optimal control problems, some extra technological constraints cause nonsmoothness. In data mining, likewise, the clustering problems have nonsmoothness. In telecommunication, determining constrained hierarchical trees for network evaluation and multicast routing cause nonsmoothness. In engineering, nonsmoothness comes from complex situations which occur when joining several bodies with corners. As for the so-called stiff problems, they are analytically smooth but numerically nonsmooth. This means that the behavior of the gradient changes unexpectedly so that these problems pretend to be nonsmooth problems. For example, it may have a similar oscillatory behavior under iterative algorithms.

1.1 Literature Review

The optimization theory, generally, can be illustrated in mathematical sense as follows,

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and X are called the objective function and the feasible set respectively. If the feasible set is $X = \mathbb{R}^n$, Problem (1.1) is referred to an unconstrained optimization problem. The general form of constrained optimization problem can be given as follows;

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h_i(x) = 0 \quad (i \in I), \\ & h_j(x) \leq 0 \quad (j \in J), \\ & x \in \mathbb{R}^n, \end{array} \tag{1.2}$$

where I and J are the index set of equality and inequality constraints respectively and $f, h_i, h_j : \mathbb{R}^n \to \mathbb{R}$ $(i \in I,\ j \in J)$. If both objective functions and constrained functions are linear functions, the problem (1.2) is called a linear optimization problem. Otherwise, it is named as a nonlinear optimization problem. As mentioned above, if at least one of the functions $f, h_i, h_j : \mathbb{R}^n \to \mathbb{R}$ $(i \in I,\ j \in J)$ is not continuously differentiable, Problem (1.2) is said to be nonsmooth.

Basically, it can be considered that the nonsmooth optimization problems consist of two types:convex and nonconvex nonsmooth problems. For the convex problems, finding global solution is easier when compared with the nonconvex problems, because every local solution is a global solution in the convex problems. Various methods have been developed to solve nonsmooth convex optimization problems, namely, the subgradient methods [1, 15, 17, 68, 71, 83, 84], different versions of the bundle methods [33, 35, 36, 49, 55, 59, 67, 82, 86] and adaptive smoothing methods [16, 25, 70, 76]. However, most of these methods do not always give efficient results for nonconvex nonsmooth problems. In real life, many practical problems are nonconvex, for examples, the area which is mentioned in the first paragraph of this chapter. The complexity of nonconvex problems arises from their nature of having multiple local solutions. Generally, most of the algorithms are able to find one of the local solutions whereas a global solution is needed. In literature, there are notable methods, namely bundle methods [34, 39, 42, 47, 58, 64, 65], discrete gradient methods [4, 14, 2], gradient sampling methods [20, 21, 23, 52], adaptive smoothing methods [75, 77, 88, 89, 90] and quasisecant methods [11, 10, 44], to solve nonconvex nonsmooth optimization problems for some special types such as locally Lipschitz continuous, lower-C^2 (i.e, the objective function is lower semi-continuous and twice times differentiable), minmax problems, etc..

Methods which have been developed for solving Problem (1.1) are usually iterative [43, 41, 83]. The idea behind the iterative methods is to obtain a sequence $\{x^k\} \in \mathbb{R}^n$ so that it can approach a local or global minimum point of Problem (1.1) by using any initial point in $\mathbb{R}^n$. The iteration is constructed by the formula $x^{k+1} = x^k + \alpha_k g^k$, where α_k and g^k are the step size and the search direction, respectively. If $f(x^{k+1}) < f(x^k)$ $(k \in \mathbb{N})$, where x^{k+1} is given as the above formula, then the direction g^k is called a *descent direction*. If the inequality holds for all k, the iterative method is called a *descent method*.

If the optimization problem is smooth, then $-\nabla f(x) \neq 0_n$ is always the steepest descent direction. In addition, if x^* is a stationary point, $\nabla f(x^*) = 0$ holds. Thus, the gradient of the objective function $\nabla f(x)$ has an important role not only to find descent directions but also to determine stopping criteria. However, in nonsmooth optimization problem, the gradients do not always exist at every points. Because of this fact, researchers need generalized gradients or other concepts such as quasidifferential, codifferential, quasisecant, discrete gradient, etc., in order to find descent direction and determine stopping criteria. Even if the gradients exist exactly at some points, they can not be useful for nonsmooth problems. In other words, "The direct applications of the gradient-based methods generally lead to failure in convergency" is emphasized in [54]. In this case, researchers use approximations via smooth functions instead of direct use of gradients of nonsmooth function or they tend to derivative free methods, such as Powell's Method [79], Nelder - Mead's method [69] and aforementioned discrete gradient methods [4, 14, 2]. Derivative free methods are untrustworthy, slow and inefficient for the large scale problems [31].

In a convex nonsmooth optimization problem, both the objective function and the constraint set are convex. Many problems possess this property both in theory and in practice. It is easy to solve these type problems both theoretically and practically [72]. A problem which satisfies the following special case of the general constrained optimization problem (1.2) is named as a convex problem:

- The objective function $f(x)$ is convex.

- The constraint set is convex. In other words:
  - the equality constraints $h_i(x)$ $(i \in I)$ are linear, and
  - the inequality constraints functions $h_j(x)$ $(j \in J)$ are concave.

If at least one of the functions $f, h_i, h_j : \mathbb{R}^n \to \mathbb{R}$ $(i \in I,\ j \in J)$ is nonsmooth, the above mentioned problem is called a convex nonsmooth optimization problem. In the nonconvex nonsmooth optimization problem, although the convex optimization theory supplies very helpful tools to nonconvex theory, finding the optimal value of the nonconvex nonsmooth optimization problem can be extremely difficult and sometimes impossible. Due to this difficulty, numerical techniques are developed, especially, including objective functions which are locally Lipschitz continuous, differences of convex functions and max-min type functions, etc..

1.1.1 Subgradient Methods

Subgradient methods developed for smooth optimization theory in the first place. In smooth theory, the most simple and understandable method is the steepest descent method, which uses the anti-gradient as a search direction:

$$d_k = -\nabla f(x_k),$$

where $\nabla f(x_k)$ is the gradient of f at the current iteration. The advantages of the steepest descent method are its low cost and easy implementation. However, its convergency is not robust because of the well-known zigzag phenomena. In order to overcome this phenomena, the conjugate gradient method have been developed in smooth theory. This method uses not only the gradient at current iteration point but also the gradient at previous iteration point. Mathematically formulated,

$$g_k = -\nabla f(x_k) - \lambda_k \nabla f(x_{k-1}),$$

where λ_k is a real scalar.

The steepest descent method and conjugate gradient method are based on the first-order Taylor's series expansion of the objective function $f(x)$. By using second order Taylor's series expansion, Newton's method, the most well-known method, have been developed. The search direction is computed as the following;

$$g_k = -\nabla^2 f(x_k)^{-1},$$

where $\nabla^2 f(x_k)$ is the Hessian of the objective function $f(x)$ at the current iteration. Newton's method is a powerful and very efficient and widely used method. On the other hand, there are two main drawbacks. One of them is that Newton's method can not sometimes converge to the solution if the starting point is too far away from the solution. The second drawback is time consuming because of the computation of inverse of Hessian at each iteration, especially, for large scale problems. As a consequence, the quasi-Newton's method is developed in order to decrease time consumption keeping its convergence rate. In quasi-Newton's method, the following search direction is used:

$$g_k = -B_k^{-1} \nabla f(x_k),$$

where B_k is an approximation of the Hessian matrix, which preserves the properties of the Hessian, such as positive definiteness and symmetry. In literature, the first quasi-Newton algorithm was proposed by W.C. Davidon in 1959. Then, Fletcher and Powell explored its mathematical properties over the next few years, and developed so called the Davidon-Fletcher-Powell formula (or DFP), which is rarely used today. The most commonly used quasi-Newton algorithms are the Symmetric Rank 1 (SR1) method and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, suggested independently by Broyden, Fletcher, Goldfarb, and Shanno, in 1970. Quasi-Newton methods are a generalization of the secant method. The difference among these updating formulas is that they maintain different properties of the matrix. Thus, the choice of which method should be used depends on the requirement of the problem. For example, SR1 method maintains the symmetry of the matrix but does not always guarantee the positive definiteness.

In nonsmooth theory, since the objective function is not smooth, a subgradient $\varepsilon_k \in \partial f(x_k)$ is used instead of the gradient $\nabla f(x_k)$. According to the properties of the objective function, one of the definitions given in Subsection (2.1) is used. The main idea behind the subgradient method is very simple and basically the generalization of the steepest descent method. However, the choice of any anti-subgradient direction may not guarantee the descent direction. The search direction in the subgradient method is as follows:

$$g_k = -\varepsilon_k / \|\varepsilon_k\|.$$

Besides the difficulty in choosing the descent direction, finding a stopping criterion can be seen as another difficulty, although the condition $0 \in \partial f(x)$ is known as the necessary condition being a minimum of the objective function. The difficulty originates from selecting an arbitrary subgradient, because a single subgradient does not contain whole information about the set of subdifferential.

The iteration of subgradient algorithm with the given starting point $x_0 \in \mathbb{R}^n$ in the solution of Problem (1.1) can be expressed as follows:

$$x_{k+1} = x_k - t_k \varepsilon_k, \tag{1.3}$$

where $\varepsilon_k \in \partial f(x_k)$ is any subgradient at the point x_k and $t_k > 0$ is the step-length. Obviously, the subgradient method uses step lengths instead of line search as they are used in the gradient methods. A choice of step size t_k is very important to avoid the line searches and to determine the stopping criterion. Although this iteration may be applied efficiently in some special cases, it has poor convergence. As a result, there have been many attempts in order to generalize quasi-Newton's methods into nonsmooth theory, such as space dilation method [83] and variable metric method [18].

1.1.2 Bundle Methods

The bundle methods have been developed in order to improve the poor convergency of the aforementioned subgradient methods. They are the most efficient methods and have lots of varieties. The central idea behind these methods is that the accumulated subgradient directions from past iterations form the quadratic subproblem and, then, a trial direction is obtained by solving this quadratic subproblem. Along the trial direction, a line search is performed to generate a serious step. Because of this procedure, they need a very large amount of memory to retain the information on the computer during implementation. Hence, it is not possible to store all information in practice. For more information and discussion, the studies [60] and [63] can be examined.

In literature, as a first bundle method, ε -steepest decent method introduced by Lemaréchal can be shown. This method is a combination of the cutting plane method [46] and conjugate subgradient method [53]. The main difficulty of this method is to determine a tolerance ε, which is the radius of the ball in which good approximation is expected. Briefly, the difficulty for the large ε is that the bundle does not approximate well and as for the small ε, is that there is a small decrease in the function value, which causes a bad convergency. Because of this difficulty, Lemaréchal developed the generalization of the cutting plane method. After that, this method was improved by Kiwiel [47]. Although Kiwiel gave two ideas, namely subgradient selection and aggregation and the restriction of the number of stored subgradient in [47], Kiwiel's method suffered from the scaling of the objective function and the uncertain numbers of line searches. All late versions of bundle method are developed to eliminate those drawbacks.

The most commonly used version of bundle methods are the proximal bundle method [49], which is based on the proximal algorithm [80], and bundle trust region method [82], which is a combination of bundle method and trust region idea. Although they are very similar, there is a difference between them in implementations when updating the search direction. As another bundle methods, the following methods can be shown: the infeasible bundle method [81], the proximal bundle method with approximate subgradient [50], the proximal-projective bundle method [51], the limited memory bundle method [37, 38] and the limited memory interior point bundle method [45].

1.1.3 Gradient Sampling Methods

Gradient sampling idea was used in [30, 83] for the first time. Later, the gradient sampling method was used to approximate the Clarke subdifferential for locally Lipschitz functions in [20] and it was improved for nonsmooth nonconvex problems in [23]. Later, other versions of gradient sampling methods for some special optimization problems was developed such as [22, 19, 52].

The locally Lipschitz functions are differentiable almost everywhere, which is proved by Rademacher's Theorem; in other words, they are not differentiable on a set of the measure zero, so the subgradient at a randomly selected point is uniquely determined as the gradient at that point. Therefore, in the gradient sampling methods, gradients are computed on a set of randomly generated nearby points at current iteration. Consequently, by using gradient sampling, a local information of the function is obtained and the quadratic subproblem is formed. The ε-steepest descent direction is constructed by solving this quadratic subproblem, where ε is the sample radius.

1.1.4 Discrete Gradient Method

The discrete gradient method uses the concept of the discrete gradient instead of the ordinary gradient or subgradient. It tries to approximate subgradient at only the final step of its algorithm, so it is different than subgradient methods. Thus, it is also known as a *derivative free method*. In [14, 13], the search direction is selected by finding the opposite of the closest point to the origin in a set of discrete gradients, which is a convex hull:

$$g_k = -v_k/||v_k||,$$

where v_k is the closest point to the origin in the convex hull of a set of discrete gradient.

1.1.5 Codifferential Methods

The need for describing the notation of codifferentiability arises due to the lack of continuity of the quasidifferential or other differential objects. Codifferentiability allows us to approximate a nonsmooth function continuously. In other words, the codifferential mapping is Hausdorff continuous for most practical classes in nonsmooth theory. The codifferential has also good differential properties, so the class of codifferentiable functions is a linear space closed in terms of what the most essential operators are. Moreover, one can explicitly give the set which consists of the elements of the codifferential for some important classes of nonsmooth functions. For the construction of the whole codifferential, some operations with polytopes are necessary; therefore, in numerical meaning, it is too complicated to form the whole set. Thus, in this thesis, truncation of the whole codifferential is used. In the literature, there are a few studies which use codifferential (see Subsection 2.3 and [26, 27, 91]) because it is considered that either the entire codifferential or its subsets should be computed at any point; however, these assumptions are too restrictive. Actually, their calculations are not possible for many class of nonsmooth functions. The methods which use codifferential are generally designed by using few elements of codifferential, such as [3, 6, 26].

1.1.6 Quasisecant Methods

The concept of secant is widely used in optimization theory, such as quasi-Newton methods. The notion of secant for locally Lipschitz functions was given in [9]. The secant method is not better than bundle methods for not only nonsmooth convex function but also nonconvex nonsmooth functions; however, it gives better results than bundle methods for nonsmooth nonconvex nonregular functions. The computation of secants is not always possible. For this reason, the notion of quasisecant was introduced by replacing strict equality in the definition of secants by inequality in [10]. Because of that, by definition it is obvious that any secant is also quasisecant but the contrary is not correct; in other words, any quasisecant is not secant. Quasisecants can be easily computed for both convex nonsmooth functions and nonconvex nonsmooth functions. The brief explanation of quasisecants will be given in Subsection 2.2. Quasisecants overestimates the objective function in some neighborhood of a given point and subgradients are used to obtain quasisecants. In literature, the quasisecant method for nonconvex nonsmooth function was firstly introduced in [10] and modified in [11, 44], where quasisecant are used to find descent direction and the idea behind it is similar to the bundle and gradient sampling methods.

1.2 Outline of the Thesis

The organization of this thesis will be as the following. Firstly, Chapter 2 contains a brief summary of theoretical background, which is about the Clarke subdifferential, quasidifferential functions, codifferentiable functions and quasisecants. The codifferentials and quasidifferentials of the special class of the functions will be located in Chapter 2. Secondly, using codifferential concept, the truncated codifferential method will be developed for convex nonsmooth unconstrained problems, which will be explained in Chapter 3. The convergence of it will be proved. In the following Chapter 4, the TCM will be advanced in order to reduce the number of gradient evaluations by using some codifferential from previous iterations. After that, a codifferential method will be developed using limited number of codifferential in Chapter 5. While we are finding the descent direction at each iteration, a fixed number codifferentials will be used. One of them includes aggregate information about previous calculated codifferential. When the number increases, this method will be more complex. If it is allowed that number is free at each iterations, this method will became the TCM. Thus, this method for the small fixed number is the simplest one among above mentioned methods. Numerical results will be obtained for the number 2, 3, 4, 12, 50 and 100, which shows us how many codifferential are used to find a descent direction in each iteration. The next Chapter 6 gives numerical results about the methods mentioned in Chapters 3, 4 and 5 by using performance profile, which will be briefly explained in Chapter 6. In that chapter, there is information about the test problems, which is used for comparison. The following Chapter 7 is just adaptation of the TCM for linearly constrained optimization problem. Using slack variables and making some calculation, linearly constrained problems will be converted to unconstrained problems. It will be proved that all properties which are needed to apply the TCM are preserved during this conversion. In the last Chapter 8, a generalized subgradient method with piecewise linear subproblem will be developed via quasisecants for locally Lipschitz problem, which is another important type of nonsmooth theory. We shall show that a set of linear inequalities must be solved to find a descent direction sufficiently. The subgradient algorithm will be used when minimizing this piecewise linear functions. In order to compare the numerical results, subgradient method will be used. The conclusion of this thesis will be given in the last part.

CHAPTER 2

THEORETICAL BACKGROUND

In this chapter, some theoretical background will be given briefly. Firstly, we will provide the subgradient for convex functions and the Clarke subdifferential for locally Lipschitz functions. Secondly, in Section 2.2, quasidifferentiability will be explained. After that, in Section 2.3, we shall give the definition of codifferentiable functions and some explanations for some classes of functions, which are useful for computational point of view. Then we shall state some basic properties of codifferentiable functions in codifferential calculus. Lastly, in Section 2.4, the concept of quasisecant will be explained, and quasisecants will be presented for some important classes of functions.

2.1 Subdifferential

In this section, definition of the subdifferential will be given for both convex and Lipschitz continuous functions as a generalization of differential. One can give all differentiable rules of subgradient as generalizations of classical differentiable rules. For example, Mean-Value Theorem, Chain Rule and Products Rule, etc., can be listed. However, inclusions have to be used instead of equalities for more details see [64].

2.1.1 Subdifferential for Convex Functions

In implementation, we will used the following subdifferential definition. In the literature, there is also $\varepsilon-$subdifferential of the convex functions and its generalization for nonconvex functions. Since they will not be used in this thesis, these definitions will not be placed.

Definition 2.1 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is called **convex** if and only if the following condition holds:*

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2) \quad \forall x_1 \text{ and } x_2 \in \mathbb{R}^n \text{ and } \forall t \in [0,1].$$

Definition 2.2 *The **subdifferential** of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ at a point x is defined by*

$$\partial f(x) = \{\zeta \in \mathbb{R}^n | \quad f(y) \geq f(x) + \langle \zeta, y - x \rangle \ \forall y \in \mathbb{R}^n\}.$$

Each vector of above mentioned set is called a subgradient of f at x. If the convex function f is continuously differentiable, then $\partial f(x) = \{\nabla f(x)\}$ by definition.

2.1.2 The Clarke Subdifferential for Locally Lipschitz Functions

Definition 2.3 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be a* ***locally Lipschitz*** *function if there exists $L > 0$ such that $\forall x, y \in \mathbb{R}^n$*

$$|f(x) - f(y)| \le L\|x - y\|,$$

where $\|\cdot\|$ is the Euclidean norm.

Clarke introduced the generalization of subdifferential for locally Lipschitz functions [24]. Using almost everywhere differentiability of locally Lipschitz functions, the Clarke subdifferential can be given as follows.

Definition 2.4 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function. The* ***Clarke subdifferential*** *of f at the point x is defined by*

$$\partial f(x) = co \left\{ v \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists\; (\, x^k \in D(f)\ (k \in \mathbb{N}), x^k \to x\ (k \to +\infty)) \\ \text{such that } v = \lim_{k \to +\infty} \nabla f(x^k) \end{array} \right\},$$

where the set $D(f)$ consists of the point at which f is differentiable, co denotes the convex hull of a set.

It is shown in [24] that "The mapping $\partial f(x)$ is upper semicontinuous and bounded on bounded sets." For locally Lipschitz functions, classical directional derivatives may not exist. Therefore, the generalized directional derivative is defined.

Definition 2.5 *The* ***generalized directional derivative*** *of $f : \mathbb{R}^n \to \mathbb{R}$ at x in the direction g is defined as*

$$f^0(x, g) = \limsup_{y \to x, \alpha \to +0} \alpha^{-1}[f(y + \alpha g) - f(y)].$$

In [2], it is reported that

"If a function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, then the generalized directional derivative exists and $f^0(x, g) = \max\{\langle v, g\rangle : v \in \partial f(x)\}$.

The function $f : \mathbb{R}^n \to \mathbb{R}$ is called a *Clarke regular* function on $\mathbb{R}^n$, if it is differentiable with respect to any direction $g \in \mathbb{R}^n$ and $f'(x, g) = f^0(x, g)$ for all $x, g \in \mathbb{R}^n$, where $f'(x, g)$ is a derivative of the function f at the point x in the direction g: $f'(x, g) = \lim_{\alpha \to +0} \alpha^{-1}[f(x + \alpha g) - f(x)]$."

Let f be a locally Lipschitz function defined on $\mathbb{R}^n$. The necessary optimality condition for the point x is

$$0 \in \partial f(x).$$

2.2 Quasidifferential

The concept of quasidifferential is a generalization of the idea of a gradient. In other words, it offers to replace the concept of a gradient in the smooth case and the concept of a subdifferential in the convex case. Quasidifferential preserves most operation of classical differential calculus (for more information see [28]). In addition to these operation, the quasidifferential allows us to find maxima and minima pointwisely.

Definition 2.6 *Let assume the function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz at the point $x \in \mathbb{R}^n$. The function f is called* ***semismooth*** *at x, if the following limit exists:*

$$\lim_{g' \to g, \alpha \to +0} \langle v, g \rangle \quad \forall v \in \partial f(x + \alpha g').$$

for every $g \in \mathbb{R}^n$.

An interesting and important nondifferentiable functions is generated by smooth compositions of semismooth functions. Thus, it should be emphasized here that the class of semismooth functions is commonly encountered in the literature. This class contains some important functions such as, convex, concave, max-type and min-type [66]. If a function f is semismooth, the directional derivative of it follows:

$$f'(x, g) = \lim_{g' \to g, \alpha \to +0} \langle v, g \rangle, \quad v \in \partial f(x + \alpha g').$$

Definition 2.7 *Assume a function f is locally Lipschitz and directionally differentiable at the point x. The function f is called* ***quasidifferentiable*** *at the point x if there exist convex, compact sets $\underline{\partial} f(x)$ and $\overline{\partial} f(x)$ such that:*

$$f'(x, g) = \max_{u \in \underline{\partial} f(x)} \langle u, g \rangle + \min_{v \in \overline{\partial} f(x)} \langle v, g \rangle.$$

The sets $\underline{\partial} f(x)$ and $\overline{\partial} f(x)$ are called a subdifferential and a superdifferential respectively. The pair of these sets $[\underline{\partial} f(x), \overline{\partial} f(x)]$ is a quasidifferential of the function f at a point x [27]. In case $\overline{\partial} f(x) = \{0\}$ (or $\underline{\partial} f(x) = \{0\}$), the function f is called subdifferentiable (or superdifferentiable). If a function is subdifferentiable, quasidifferential and subdifferential are coincident.

2.2.1 Quasidifferential of Smooth Functions

Assume f is continuously differentiable in some neighborhood of a point $x \in X \subset \mathbb{R}^n$. Obviously, f is quasidifferentiable at x and the following pairs are quasidifferentials of f:

$$[\{\nabla f(x)\}, \{0\}] \text{ and } [\{0\}, \{\nabla f(x)\}].$$

Thus, a smooth functions is both subdifferentiable and superdifferentiable.

2.2.2 Quasidifferential of Convex Functions

Suppose a function f is convex defined on an open set $X \subset \mathbb{R}^n$. Because of the fact that directionally differentiability of f, the quasidifferential of f can be given as follows:

$$[\underline{\partial} f(x), \{0\}],$$

where $\underline{\partial} f(x) = \partial f(x) = \{\zeta | \zeta \in \mathbb{R}^n, \ f(y) \geq f(x) + \langle \zeta, y - x \rangle \ \forall y \in \mathbb{R}^n\}$ is the subdifferential of f at the point x.

2.2.3 Quasidifferential of Concave Functions

Under the assumption being concave on the function f, which is defined on an open set $X \subset \mathbb{R}^n$, analogously, the quasidifferential of f can be given as follows:

$$[\{0\}, \overline{\partial} f(x)],$$

where $\overline{\partial} f(x) = \partial f(x) = \{\zeta | \zeta \in \mathbb{R}^n,\ f(y) \geq f(x) + \langle \zeta, y - x \rangle\ \forall y \in \mathbb{R}^n\}$ is the subdifferential of f at the point x.

2.3 Codifferential

The lack of continuity of the subdifferential and quasidifferential mappings causes difficulties in the study of optimization theory. In [86], it was noted that the lack of this property was responsible for the failure of nonsmooth steepest descent algorithms. On the other hand, the codifferential mapping for the convex functions is Hausdorff continuous. Thus, for developing optimization methods in thesis, it will be used mostly.

Definition 2.8 *Let X be an open subset of* $\mathbb{R}^n$. *Assume that co* $\{x, x + \Delta\} \subset X$. *A function* $f : X \to \mathbb{R}$ *is called* ***codifferentiable*** *at the point* $x \in X$ *if there exists a pair* $Df(x) = \left[\underline{d}f(x), \overline{d}f(x)\right]$, *where the sets* $\underline{d}f(x)$ *and* $\overline{d}f(x)$ *are convex compact sets in* $\mathbb{R}^{n+1}$, *such that*

$$f(x + \Delta) = f(x) + \max_{(a,v) \in \underline{d}f(x)} [a + \langle v, \Delta \rangle] + \min_{(b,u) \in \overline{d}f(x)} [b + \langle u, \Delta \rangle] + o_x(\Delta), \tag{2.1}$$

where

$$\frac{o_x(\alpha\Delta)}{\alpha} \to 0 \text{ as } \alpha \downarrow 0 \text{ for all } \Delta \in \mathbb{R}^n \tag{2.2}$$

and

$$a, b \in \mathbb{R} \quad v, w \in \mathbb{R}^n.$$

The pair $Df(x) = \left[\underline{d}f(x), \overline{d}f(x)\right]$ is called a *codifferential* of the function f at the point x, the sets $\underline{d}f(x)$ and $\overline{d}f(x)$ are called *hypodifferential* and *hyperdifferential*, respectively. Elements of their are called hypogradients and hypergradients respectively. Note that the codifferential is not unique [27].

If a function f is codifferentiable in some neighborhood of a point x, f is called *codifferentiable* and the mapping Df is called *codifferential*.

A function f is called *uniformly codifferentiable* at a point x in directions, if (2.2) holds uniformly in $S_1 = \{\Delta \in \mathbb{R}^n \mid \ \|\Delta\| = 1\}$.

A function f is called *continuously codifferentiable* at a point x, if it is codifferentiable in some neighborhood of the point x and the mapping Df is Hausdorff continuous at x.

If $\overline{d}f(x) = \{0_{n+1}\}$ (or $\underline{d}f(x) = \{0_{n+1}\}$), the function f is called *hypodifferentiable* (or *hyperdifferentiable*), where 0_{n+1} denotes the zero element of the space $\mathbb{R}^{n+1}$.

With respect to computation, the class of the codifferentiable function whose hypodifferential and hyperdifferential are polyhedral, i.e, convex hulls of a finite number of points, is useful [27]. The following functions are in that class (for more functions class and explanations, see [27]).

2.3.1 Codifferential of Smooth Functions

Let f be continuously differentiable in some neighborhood of a point $x \in X \subset \mathbb{R}^n$. Then

$$f(x+\Delta) = f(x) + \langle \nabla f(x), \Delta \rangle + o_x(\Delta), \tag{2.3}$$

where $\frac{o_x(\Delta)}{\|\Delta\|} \to 0$ as $\|\Delta\| \to 0$, $\nabla f(x)$ is the gradient of the function f at the point x. Now, consider the following sets:

$$\begin{aligned} \underline{d}f(x) &= \{(0, \nabla f(x))\} \subset \mathbb{R}^{n+1}, \\ \overline{d}f(x) &= \{0_{n+1}\} \subset \mathbb{R}^{n+1}. \end{aligned}$$

By (2.3), we obtain

$$\begin{aligned} f(x+\Delta) &= f(x) + 0 + \langle \nabla f(x), \Delta \rangle + 0 + \langle 0, \Delta \rangle + o_x(\Delta) \\ &= f(x) + \max_{(a,v)\in \underline{d}f(x)} [a + \langle v, \Delta \rangle] + \min_{(b,u)\in \overline{d}f(x)} [b + \langle u, \Delta \rangle] + o_x(\Delta), \end{aligned}$$

where $\underline{d}f(x)$ and $\overline{d}f(x)$ are as introduced above.

Thus, f is codifferentiable with $Df(x) = [\{(0, f'(x))\}, \{0_{n+1}\}]$. In addition, f is continuously codifferentiable in a neighbourhood of the point x uniformly in directions [27].

As a codifferential, the following pair can be also chosen

$$Df(x) = [\{0_{n+1}\}, \{(0, \nabla f(x))\}].$$

As a result, the function f is both hypodifferentiable and hyperdifferentiable and even continuously hypodifferentiable and hyperdifferentiable. In this example, it can be observed that the codifferential is not unique.

2.3.2 Codifferential of Convex Functions

Let a function f be convex and finite on $X \subset \mathbb{R}^n$, $U \subset X$ be a closed bounded set and $x \in \operatorname{int} U$. From the definition of the subgradient, we have the following inequality at the point x:

$$f(x) \geq f(z) + \langle v_z, x - z \rangle,$$

where $v_z \in \partial f(z)$, $\forall z \in U$ and

$$f(x) = \max_{z\in U} \{f(z) + \langle v_z, x - z \rangle\}.$$

At a point $x + \Delta \in \operatorname{int} U$ we have

$$\begin{aligned} f(x+\Delta) &= f(x) + \max_{z\in U} \{f(z) - f(x) + \langle v_z, x + \Delta - z \rangle\} \\ &= f(x) + \max_{(a,v)\in \underline{d}f(x)} \{a + \langle v, \Delta \rangle\}, \end{aligned}$$

where the set $\underline{d}f(x)$ is the hypodifferential of the function f at the point x. The set is defined as follows [27, 91]:

$$\underline{d}f(x) = \operatorname{cl}\ \operatorname{co} \{(a, v) \in \mathbb{R} \times \mathbb{R}^n : a = f(z) - f(x) + \langle v, x - z \rangle,\ v \in \partial f(z),\ \forall\ z \in U\}. \tag{2.4}$$

Thus, the codifferential of a convex function is the pair $Df(x) = \left[\underline{d}f(x), \overline{d}f(x)\right]$, where $\underline{d}f(x)$ is as in (2.4) and $\overline{d}f(x) = \{0_{n+1}\}$, so convex functions are continuously hypodifferentiable[27].

2.3.3 Codifferential of Concave Functions

Let a function f be concave and finite on $X \subset \mathbb{R}^n$, $U \subset X$ be a closed bounded set and $x \in \text{int}U$. Let f be expressed as $-g$, where g is a convex function.

The definition of the supergradient of the function f at the point x implies that

$$g(x) \geq g(z) + \langle w_z, x - z \rangle,$$

where $w_z \in \partial g(z)\ \forall z \in U$.

By using the same idea in Subsection (2.3.2), the following results are obtained.

The codifferential of a concave function is the pair $Df(x) = \left[\underline{d}f(x), \overline{d}f(x)\right]$, where

$$\begin{aligned}&\underline{d}f(x) = \{0_{n+1}\} \text{ and}\\ &\overline{d}f(x) = \text{cl co}\{(b, u) \in \mathbb{R} \times \mathbb{R}^n \mid b = -f(z) + f(x) - \langle u, x - z \rangle\ u \in \partial f(z),\ \forall\ z \in U\}.\end{aligned}$$

Concave functions are continuously hypedifferential.

2.3.4 Codifferential of Difference of Two Convex Functions

Let f be a d.c. (i.e, difference of two convex functions) function and a closed bounded set $U \subset \mathbb{R}^n$, a point $x \in \text{int}U$. f is expressed in the following form:

$$f(x) = p(x) - q(x),$$

where $p, q : \mathbb{R}^n \to \mathbb{R}$ are convex.

For any $z \in U$ take subgradients $v_z \in \partial p(x)$ and $u_z \in \partial q(x)$. The subgradient of the function f at the point x implies the following inequality:

$$\begin{aligned}p(x) &\geq p(z) + \langle v_z, x - z \rangle \quad \forall z \in U,\\ q(x) &\geq q(z) + \langle u_z, x - z \rangle \quad \forall z \in U;\end{aligned}$$

so

$$\begin{aligned}f(x) &= p(x) - q(x) = \max_{z \in U}\{p(z) + \langle v_z, x - z \rangle\} - \max_{z \in U}\{q(z) + \langle u_z, x - z \rangle\}\\ &= \max_{z \in U}\{p(z) + \langle v_z, x - z \rangle\} + \min_{z \in U}\{-q(z) - \langle u_z, x - z \rangle\}.\end{aligned}$$

At the point $x + d \in U$ we have

$$\begin{aligned}& f(x+d) &&= \max_{z \in U}\{p(z) + \langle v_z, x + d - z \rangle\} + \min_{z \in U}\{-q(z) - \langle u_z, x + d - z \rangle\}\\ \Rightarrow\ & f(x+d) - f(x) &&= \max_{z \in U}\{p(z) - p(x) + \langle v_z, x + d - z \rangle\} + \min_{z \in U}\{-q(z) + q(x) - \langle u_z, x + d - z \rangle\}\\ \Rightarrow\ & f(x+d) - f(x) &&= \max_{(a,v) \in \underline{d}f(x)}\{a + \langle v, d \rangle\} + \min_{(b,u) \in \overline{d}f(x)}\{b + \langle -u, d \rangle\},\end{aligned}$$

where $\underline{d}f(x)$ is hypodifferential of f at the point x and $\overline{d}f(x)$ is hyperdifferential of f at the point x and they are given as the following

$$\begin{aligned}\underline{d}f(x) &= \text{cl co}\{(a, v) \in \mathbb{R} \times \mathbb{R}^n \mid a = p(z) - p(x) + \langle v, x - z \rangle,\ v \in \partial p(z),\ \forall\ z \in U\}\\ \overline{d}f(x) &= \text{cl co}\{(b, -u) \in \mathbb{R} \times \mathbb{R}^n \mid b = -q(z) + q(x) - \langle u, x - z \rangle,\ u \in \partial q(z),\ \forall\ z \in U\},\end{aligned}$$

where co and cl denote convex hull and closure respectively.

2.3.5 Properties of Codifferentiable Functions

In this section, it is assumed that f is defined on an open subset X of $\mathbb{R}^n$ and co $\{x, x+\Delta\} \subset X$. The proofs of the all properties can be found in [27].

Lemma 2.9 *Let f_i $(i = 1, 2, ..., N)$ be codifferentiable (continuously codifferentiable) at a point $x \in X$, then the function $f = \sum_{i=1}^{N} c_i f_i$ with real coefficients c_i $(i = 1, 2, ..., N)$ is also codifferentiable (continuously codifferentiable) at x and its codifferential is the following set:*

$$Df(x) = \sum_{i=1}^{N} c_i Df_i(x), \tag{2.5}$$

where $Df_i((x) = \left[\underline{d}f_i(x), \overline{d}f_i(x)\right]$ is a codifferential of the function f_i at x $(i = 1, 2, ..., N)$.

Remark 2.10 *The above mentioned codifferential of the function f is only one of codifferentials of it.*

Lemma 2.11 *Let f_1 and f_2 be codifferentiable (continuously codifferentiable) at the point $x \in X$. The function $f = f_1 f_2$ is also codifferentiable (continuously codifferentiable) at x and its codifferential is the following set:*

$$Df(x) = f_1(x)Df_2(x) + f_2(x)Df_1(x). \tag{2.6}$$

In addition, if the functions f_1 and f_2 are codifferentiable uniformly in directions, then f is also codifferential uniformly in directions [27].

Lemma 2.12 *Let a function f_1 be codifferentiable (continuously codifferentiable) at a point $x \in X$ and $f_1(x) \neq 0$. The function $f = \dfrac{1}{f_1}$ is codifferentiable (continuously codifferentiable) at the poit x and its codifferential is the following*

$$Df(x) = -\frac{1}{f_1^2(x)} Df_1(x). \tag{2.7}$$

Lemma 2.13 *Let functions φ_i for $i = 1, 2, ..., N$ be codifferentiable (continuously codifferentiable) at a point $x \in X$. The functions $f_1(y) = \max_{i=1,2,...,N} \varphi_i(y)$ and $f_2(y) = \min_{i=1,2,...,N} \varphi_i(y)$ are also codifferentiable at x and their codifferentials are $Df_1(x) = \left[\underline{d}f_1(x), \overline{d}f_1(x)\right]$ and $Df_2(x) = \left[\underline{d}f_2(x), \overline{d}f_2(x)\right]$, where*

$$\underline{d}f_1(x) = co \left\{ \underline{d}\varphi_k(x) - \sum_{\substack{i=1 \\ i \neq k}}^{N} \overline{d}\varphi_i(x) + \{(\varphi_k(x) - f_1(x), 0_n)\} \mid k = 1, 2, ..., N \right\}, \tag{2.8}$$

$$\overline{d}f_1(x) = \sum_{i=1}^{N} \overline{d}\varphi_i(x), \quad \underline{d}f_1(x) = \sum_{i=1}^{N} \underline{d}\varphi_i(x), \tag{2.9}$$

$$\overline{d}f_2(x) = co \left\{ \overline{d}\varphi_k(x) - \sum_{\substack{i=1 \\ i \neq k}}^{N} \underline{d}\varphi_i(x) + \{(\varphi_k(x) - f_2(x), 0_n)\} \mid k = 1, 2, ..., N \right\}. \tag{2.10}$$

Hence, the class of codifferentiable (respectively continuously codifferentiable) functions is a linear space closed with respect to all smooth operations and with respect to the operations of taking the pointwise maximum and minimum over a finite number of points [27].

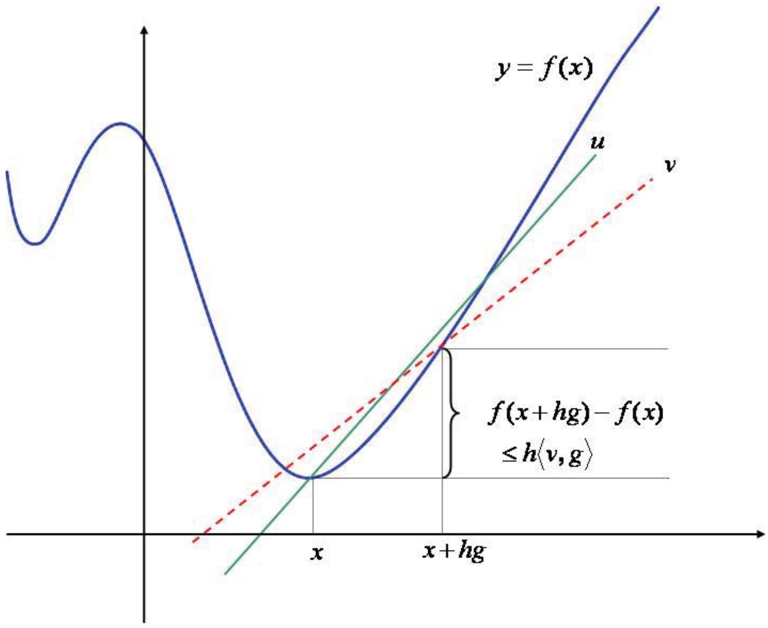


Figure 2.1: Quasisecants for a one variable function

2.4 Quasisecants

The concept of secants is commonly used in not only smooth optimization but also nonsmooth optimization theory. For instance, secants have been used in quasi-Newton methods. In this section, the definition of quasisecants for locally Lipschitz functions is given.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz function and $h > 0$ be a given real number.

Definition 2.14 *A **quasisecant** v of the function f at the point x is a vector in $\mathbb{R}^n$. It depends on the selection of the direction $g \in S_1$ and the length $h > 0$. According to the direction and the length, the quasisecant is defined as follows:*

$$f(x + hg) - f(x) \leq h\langle v, g\rangle.$$

Figure 2.1 presents examples of quasisecants in univariate case.

The notation $v(x, g, h)$ is used for any quasisecant at the point x in the direction $g \in S_1$ with the length $h > 0$ corresponding function f.

The set of quasisecants of the function f at a point x is given as follows for fixed $h > 0$:

$$QSec(x, h) = \{w \in \mathbb{R}^n : \ \exists(g \in S_1), \ \ w = v(x, g, h)\}.$$

When $h \downarrow 0$, the set consists of limit points of quasisecants can be given as follows:

$$QSL(x) = \left\{w \in \mathbb{R}^n : \ \exists(g \in S_1, \{h_k\}) : \ h_k > 0, \ \lim_{k\to\infty} h_k = 0 \text{ and } w = \lim_{k\to\infty} v(x, g, h_k)\right\}.$$

A mapping $x \mapsto QSec(x,h)$ is called *a subgradient-related (SR)-quasisecant mapping* if the corresponding set $QSL(x) \subseteq \partial f(x)$ for all $x \in \mathbb{R}^n$. In this case, elements of $QSec(x,h)$ are called *SR-quasisecants*. In Subsections 2.4.1-2.4.4 and in Chapter 8, SR-quasisecants are used. In the following sections, SR-quasisecants are presented for some classes of functions.

2.4.1 Quasisecants of Smooth Functions

Assume that the function f is continuously differentiable. Then,

$$v(x,g,h) = \nabla f(x+hg) + \alpha g \ (g \in S_1,\ h > 0)$$

is a quasisecant at a point x with respect to the direction $g \in S_1$. Here,

$$\alpha = \frac{f(x+hg) - f(x) - h\langle \nabla f(x+hg), g\rangle}{h}.$$

Obviously, $v(x,g,h) \to \nabla f(x)$ as $h \downarrow 0$. As a conclusion, each $v(x,g,h)$ is SR-quasisecant at the point x.

2.4.2 Quasisecants of Convex Functions

Assume that the function f is proper convex, in other words takes any real value for any point x, bounded below and convex. Since

$$f(x+hg) - f(x) \le h\langle v, g\rangle \ \ \forall v \in \partial f(x+hg),$$

any $v \in \partial f(x+hg)$ is a quasisecant at the point x. Then we have

$$QSec(x,h) = \bigcup_{g\in S_1} \partial f(x+hg).$$

Since the sundifferential map is the upper semicontinuous, the set $QSL(x)$ is subset of the subdifferential $\partial f(x)$. This allows us to calculate a SR-quasisecant v at the point x as $v \in \partial f(x+hg)$.

2.4.3 Quasisecants of Maximum Functions

Consider the following function, which is maximum of some locally Lipschitzian functions f_i $(i = 1,\dots,m)$:

$$f(x) = \max_{i=1,\dots,m} f_i(x).$$

Consider the following set for any $g \in S_1$:

$$R(x+hg) = \{i \in \{1,\dots,m\} \mid \ f_i(x+hg) = f(x+hg)\}.$$

The set $QSec(x,h)$ of quasisecants at a point x is defined as

$$QSec(x,h) = \bigcup_{g\in S_1} \left\{ v^i(x,g,h) \mid i \in R(x+hg) \right\}.$$

where $v^i \in \mathbb{R}^n$ is a SR-quasisecant of the function f_i at a point x. Since the subdifferential map is an upper semicontinuous map, the set $QSL(x)$ is a subset of the subdifferential $\partial f(x)$. SR-quasisecants of the function f are defined as above.

2.4.4 Quasisecants of D.C. Functions

In this subsection, the differences of two convex function is examined, mathematically it can be given as follows:

$$f(x) = f_1(x) - f_2(x),$$

where functions f_1 and f_2 are convex functions.

A quasisecant of the function f at the point x can be computed as $v = v^1 - v^2$ where subgradients $v^1 \in \partial f_1(x + hg),\ v^2 \in \partial f_2(x)$.

On the other hand, aforementioned quasisecants does not need to be SR-quasisecants. As reported in [10]:

"Since d.c. functions are quasidifferentiable [27] and if additionally subdifferentials $\partial f_1(x)$ and $\partial f_2(x)$ are polytopes, one can use an algorithm from [14, 13] to compute subgradients v^1 and v^2 such that their difference will converge to a subgradient of the function f at the point x."

As a result, this algorithm can be used to compute SR-quasisecants of the function f at the point x. Subdifferential and superdifferential of d.c. functions can be given as follows:

$$F_1(x) = \sum_{i=1}^{m} \min_{j=1,\dots,p} f_{ij}(x),$$

$$F_2(x) = \max_{i=1,\dots,m} \min_{j=1,\dots,p} f_{ij}(x).$$

Here, functions f_{ij} are continuously differentiable and proper convex. SR-quasisecants satisfy the following condition: for any $\varepsilon > 0$ there exists $\delta > 0$ such that

$$QSec(y, h) \subset \partial f(x) + B_\varepsilon(0) \tag{2.11}$$

for all $x \in B_\delta(x)$ and $h \in (0, \delta)$. This is always true for functions considered above.

CHAPTER 3

TRUNCATED CODIFFERENTIAL METHOD

In this chapter, a new algorithm to minimize convex functions will be developed. This algorithm will be based on the concept of codifferential. Since the computation of whole codifferential is not always possible we shall propose an algorithm for computation of descent directions using only a few elements from the codifferential. The convergence of the proposed minimization algorithm will be proved and results of numerical experiments using a set of test problems with not only nonsmooth convex but also nonsmooth nonconvex objective function will be reported in Chapter 6 by comparing the proposed algorithm with some other algorithms.

3.1 Introduction

In this section we focus on solving the following problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array} \tag{3.1}$$

where the objective function f is assumed to be proper convex. In the literature, there are several numerical techniques in order to solve Problem (3.1). As important techniques, subgradient methods [83], different versions of the bundle methods [33, 34, 35, 36, 39, 42, 47, 49, 55, 58, 59, 64, 65, 67, 86] can be counted. In this chapter, we propose a method, namely the truncated codifferential method for solving Problem (3.1). The notion of codifferential was firstly given in [27]. The codifferential mapping for some important classes of functions encountered in nonsmooth theory is Hausdorff continuous. In the literature, there are only a few algorithms based on the codifferential (see [26, 27, 91]), whereas the codifferential map has good differential properties. In these algorithms, it is assumed the need of the whole set of codifferentials (or its subsets). Because of this assumption, researchers did not reach the success to develop methods for many classes of nonsmooth optimization problems. In this chapter, we will show that it is actually not necessary to use the whole set of codifferential.

In this chapter, a new codifferential method is proposed for solving Problem (3.1). At each iteration of this method, just a few elements from the set of codifferentials are used to find search directions. Therefore we call this method a *truncated codifferential method*. By using these search directions, a sequence of the points is generated iteratively. It is proved that the accumulation point of this sequence is a solution of Problem (3.1). Results of numerical experiments using a set of well-known nonsmooth optimization academic test problems are reported. after that, these numerical results are used in the comparison to the proposed algorithm with the bundle method.

This chapter is structured as follows: An algorithm for finding descent directions is presented in Section 3.2. A truncated codifferential method is introduced and its convergence is examined in Section

3.3. Results of numerical experiments are visualized by using performance profiles in Section 6.3.

3.2 Computation of a Descent Direction

For the computation of search directions, a subset of the hypodifferential will be defined. It will be show that this subset is sufficient to find descent directions. For the given $\lambda \in (0,1)$ and consider the following set:

$$H(x,\lambda) = cl\ co \left\{ \quad w = (a,v) \in \mathbb{R} \times \mathbb{R}^n \quad \middle| \quad \begin{array}{l} \exists\, y \in B_\lambda(x), \\ v \in \partial f(y), \\ a = f(y) - f(x) - \langle v, y - x \rangle \end{array} \right\}. \tag{3.2}$$

It can be easily observed that $a \leq 0$ for all $w = (a,v) \in H(x,\lambda)$. Since $a = 0$ at the point x, we can conclude the following equality:

$$\max_{w=(a,v)\in H(x,\lambda)} a = 0. \tag{3.3}$$

If $\bar{B}_\lambda(x) \subset \mathrm{int}U$ for all $\lambda \in (0,1)$ where $U \subset \mathbb{R}^n$ is a closed convex set, then from the definition of both the hypodifferential and the set $H(x,\lambda)$, the following inclusion holds:

$$H(x,\lambda) \subset \underline{d}f(x) \ \ \forall\, \lambda \in (0,1).$$

The sets $H(x,\lambda)$ is called *truncated codifferentials* of the function f at the point x.

Proposition 3.1 *Assume that* $0_{n+1} \notin H(x,\lambda)$ *for a given* $\lambda \in (0,1)$ *and*

$$\|w^0\| = \min\{\|w\| : \ w \in H(x,\lambda)\} > 0, \ \textit{with} \ \ w^0 = (a_0, v^0), \tag{3.4}$$

where $\|\cdot\|$ *denotes the Euclidean norm. Then,* $v^0 \neq 0_n$ *and*

$$f(x + \lambda g^0) - f(x) \leq -\lambda\|v^0\|, \tag{3.5}$$

where $g^0 = -\|w^0\|^{-1}v^0$.

Proof: Since w^0 is a solution of 3.4,

$$\langle w^0, w - w^0 \rangle \geq 0 \quad \forall w = (a,v) \in H(x,\lambda)$$

or

$$a_0 a + \langle v^0, v \rangle \geq \|w^0\|^2. \tag{3.6}$$

First, $v^0 \neq 0_n$ should be proved. Assume that $v^0 = 0_n$. Since $w \neq 0_{n+1}$ we get that $a_0 < 0$. Then, it follows from (3.6) that $a_0(a - a_0) \geq 0$ or $a \leq a_0 < 0$. In other words, $a < 0$ for all $w = (a,v) \in H(x,\lambda)$, which contradicts (3.3).

Now we will prove (3.5). Dividing both sides of (3.6) by $-\|w^0\|$, we obtain

$$-\frac{a_0 a}{\|w^0\|} + \langle v, g^0 \rangle \leq -\|w^0\|. \tag{3.7}$$

It is clear that $\|w^0\|^{-1}a_0 \in (-1,0)$ and, since $\lambda \in (0,1)$,

$$\mu = -\frac{\lambda a_0}{\|w^0\|} \in (0,1).$$

Therefore, combining $a \le 0$ and (3.7), we get

$$a + \lambda\langle v, g^0\rangle \le \mu a + \lambda\langle v, g^0\rangle = -\frac{\lambda a_0}{\|w^0\|} a + \lambda\langle v, g^0\rangle \le -\lambda\|w^0\|. \tag{3.8}$$

Obviously, $x + \lambda g^0 \in B(x, \lambda)$. As a result, from the definition of the set $H(x, \lambda)$

$$f(x + \lambda g^0) - f(x) = a + \lambda\langle v, g^0\rangle,$$

where $w = (a, v) \in H(x, \lambda)$ and $a = f(x + \lambda g^0) - f(x) - \lambda\langle v, g^0\rangle, \;\; v \in \partial f(x + \lambda g^0)$. Then, the proof follows from (3.8). $\triangle$

According to Proposition 3.1, the truncated codifferential $H(x, \lambda)$ can be used to find descent directions of a function f. Moreover, for any $\lambda \in (0, 1)$, the truncated codifferential can be used in the calculation of descent directions. Unfortunately, it is generally not possible to find descent direction by using Proposition 3.1 since the entire set $H(x, \lambda)$ must be used. Actually, the usage of the entire set $H(x, \lambda)$ is not always possible. However, Proposition 3.1 helps us how an algorithm for finding descent directions can be developed. In order to over come this difficulty, the following algorithm is developed using only a few elements from $H(x, \lambda)$ to compute descent directions.

Let the numbers $\lambda, \; c \in (0, 1)$ and a small enough number $\delta > 0$ be given.

Algorithm 3.2 *Computation of descent directions.*

Step 1. Select any $g^1 \in S_1$, and compute $v^1 \in \partial f(x + \lambda g^1)$ and $a_1 = f(x + \lambda g^1) - f(x) - \lambda\langle v^1, g^1\rangle$. Set $\bar{H}_1(x) = \{w^1 = (a_1, v^1)\}$ and $k = 1$.

Step 2. Compute the $\bar{w}^k = (\bar{a}_k, \bar{v}^k) \in \mathbb{R} \times \mathbb{R}^n$ as a solution of the following problem:

$$\min \;\; \|w\|^2 \quad \text{subject to} \;\; w \in \bar{H}_k(x). \tag{3.9}$$

Step 3. If

$$\|\bar{w}^k\| \le \delta, \tag{3.10}$$

then **stop**. Otherwise, compute $\bar{g}^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ and go to Step 4.

Step 4. If

$$f(x + \lambda\bar{g}^k) - f(x) \le -c\lambda\|\bar{w}^k\|, \tag{3.11}$$

then **stop**. Otherwise, set $g^{k+1} = \bar{g}^k$ and go to Step 5.

Step 5. Compute $v^{k+1} \in \partial f(x + \lambda g^{k+1})$ and $a_{k+1} = f(x + \lambda g^{k+1}) - f(x) - \lambda\langle v^{k+1}, g^{k+1}\rangle$. Construct the set $\bar{H}_{k+1}(x) = \text{co}\,\{\bar{H}_k(x) \bigcup \{w^{k+1} = (a_{k+1}, v^{k+1})\}\}$, set $k \leftarrow k + 1$ and go to Step 2.

Some explanations on Algorithm 3.2 as follows. In Step 1, we compute the element of the truncated codifferential using any direction $g^1 \in S_1$. The closest point to the origin in the set of all computed codifferential is computed in Step 2. This problem is a quadratic optimization problem. In the literature there are several algorithms [32, 48, 73, 74, 87] to solve this problem. In the implementation the algorithm from [87] is used. If the norm of the closest point is less than a given tolerance $\delta > 0$, then the point x is a stationary point with tolerance $\delta > 0$; otherwise, a new search direction is computed in Step 3. If this new search direction is a descent direction, then the algorithm terminates in Step 4. Otherwise, a new codifferential in the current search direction is computed in Step 5.

Algorithm 3.2 have some similarities with respect to calculation of search direction in the bundle-type algorithms. Especially, Algorithm 3.2 is similar to the algorithm proposed in [86]. However, in Algorithm 3.2, elements of the truncated codifferential are used instead of subgradients.

In the next proposition, It is proved that Algorithm 3.2 terminates after finite number of repetitions. A standard technique is used to prove it.

Proposition 3.3 *Assume that f is proper convex function, $\lambda \in (0,1)$ and there exists $K \in (0,\infty)$ such that*

$$\max\left\{\|w\| \mid w \in \underline{d}f(x)\right\} \leq K.$$

For the any value of $c \in (0,1)$ and $\delta \in (0,K)$, Algorithm 3.2 terminates after at most m steps such that

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 2, \quad K_1 = 1 - [(1-c)(2K)^{-1}\delta]^2.$$

Proof: Since at a point x for a given $\lambda \in (0,1)$

$$\bar{H}_k(x) \subset H(x,\lambda) \subset \underline{d}f(x)$$

for any $k \in \mathbb{N}$, it follows that

$$\max\left\{\|w\| \mid w \in \bar{H}_k(x)\right\} \leq K \quad \forall k \in \mathbb{N}. \tag{3.12}$$

First, we will show that if neither stopping criteria (3.10) and (3.11) are satisfied, then a new hypogradient w^{k+1} computed in Step 5 does not belong to the set $\bar{H}_k(x)$. Assume $\bar{H}_k(x)$ belongs to w^{k+1}. Since both stopping criteria are not satisfied, it follows that $\|\bar{w}^k\| > \delta$ and

$$f(x+\lambda g^{k+1}) - f(x) > -c\lambda\|\bar{w}^k\|.$$

The definition of the hypogradient $w^{k+1} = (a_{k+1}, v^{k+1})$ implies that

$$f(x+\lambda g^{k+1}) - f(x) = a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle,$$

and we have

$$-c\lambda\|\bar{w}^k\| < a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle.$$

Putting $g^{k+1} = -\|\bar{w}^k\|^{-1}\bar{v}^k$ we get

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda}a_{k+1} < c\|\bar{w}^k\|^2. \tag{3.13}$$

Since $\bar{w}^k = \operatorname{argmin}\{\|w\|^2 : w \in \bar{H}_k(x)\}$,

$$\langle \bar{w}^k, w\rangle \geq \|\bar{w}^k\|^2$$

for all $w \in \bar{H}_k(x)$. By assumption $w^{k+1} \in \bar{H}_k(x)$, we obtain

$$a_{k+1}\bar{a}_k + \langle \bar{v}^k, v^{k+1}\rangle \geq \|\bar{w}^k\|^2. \tag{3.14}$$

Notice that $a_{k+1} \leq 0$ and $\bar{a}_k \geq -\|\bar{w}^k\|$. Then, we have $\bar{a}_k a_{k+1} \leq -\|\bar{w}^k\|a_{k+1}$. Combining this with (3.14), we obtain

$$\langle v^{k+1}, \bar{v}^k\rangle - \|\bar{w}^k\|a_{k+1} \geq \|\bar{w}^k\|^2.$$

Finally, since $\lambda \in (0, 1)$ we get

$$\langle v^{k+1}, \bar{v}^k \rangle - \frac{\|\bar{w}^k\|}{\lambda} a_{k+1} \geq \|\bar{w}^k\|^2,$$

which contradicts (3.13). Thus, if both stopping criteria (3.10) and (3.11) are not satisfied, then the new hypogradient w^{k+1} makes improvement in order to approximate to set $H(x, \lambda)$.

Obviously, $\|\bar{w}^{k+1}\|^2 \leq \|tw^{k+1} + (1 - t)\bar{w}^k\|^2$ for all $t \in [0, 1]$, which means

$$\|\bar{w}^{k+1}\|^2 \leq \|\bar{w}^k\|^2 + 2t\langle \bar{w}^k, w^{k+1} - \bar{w}^k \rangle + t^2\|w^{k+1} - \bar{w}^k\|^2.$$

Inequality (3.12) implies that

$$\|w^{k+1} - \bar{w}^k\| \leq 2K.$$

It follows from (3.13) that

$$\begin{aligned} \langle \bar{w}^k, w^{k+1} \rangle &= \bar{a}_k a_{k+1} + \langle \bar{v}^k, v^{k+1} \rangle \\ &\leq -\frac{\|\bar{w}^k\|}{\lambda} a_{k+1} + \langle \bar{v}^k, v^{k+1} \rangle \\ &\leq -c\|\bar{w}^k\|^2. \end{aligned}$$

Then, we get

$$\|\bar{w}^{k+1}\|^2 \leq \|\bar{w}^k\|^2 - 2t(1 - c)\|\bar{w}\|^2 + 4t^2 K^2.$$

Let $t_0 = (1 - c)(2K)^{-2}\|\bar{w}^k\|^2$. It is clear that $t_0 \in (0, 1)$ and, therefore,

$$\|\bar{w}^{k+1}\|^2 \leq \left\{1 - [(1 - c)(2K)^{-1}\|\bar{w}^k\|]^2\right\} \|\bar{w}^k\|^2. \tag{3.15}$$

Since $\|\bar{w}^k\| > \delta$ for all $k = 1, \ldots, m - 1$, it follows from (3.15) that

$$\|\bar{w}^{k+1}\|^2 \leq \{1 - [(1 - c_1)(2K)^{-1}\delta]^2\}\|\bar{w}^k\|^2.$$

Let $K_1 = 1 - [(1 - c_1)(2K)^{-1}\delta]^2$. Then, $K_1 \in (0, 1)$ and we have

$$\|\bar{w}^m\|^2 \leq K_1\|\bar{w}^{m-1}\|^2 \leq \ldots \leq K_1^{m-1}\|\bar{w}^1\|^2 \leq K_1^{m-1}K^2.$$

Thus, the inequality $\|\overline{w}\| \leq \delta$ is satisfied if $K_1^{m-1}K^2 \leq \delta^2$. This inequality must happen after at most m steps, where

$$m \leq 2 \log_2(\delta/K)/ \log_2 K_1 + 2.$$

△

Definition 3.4 *A point $x \in \mathbb{R}^n$ is called a (λ, δ)-stationary point of the function f if*

$$\min_{w \in H(x,\lambda)} \|w\| \leq \delta.$$

It can be easily observed that Algorithm 3.2 for a point x either finds a descent direction or determines the point x as a (λ, δ)-stationary point for the convex function f.

3.3 A Truncated Codifferential Method

In this section, the truncated codifferential method to find the solution of problem (3.1) is introduced. First of all, we should find stationary points with some tolerance. According to this purpose, the following algorithm was designed to find (λ, δ)-stationary points for given numbers $\lambda \in (0, 1),\ c_1 \in (0, 1), c_2 \in (0, c_1]$ and the tolerance $\delta > 0$.

Algorithm 3.5 *The truncated codifferential method for finding (λ, δ)-stationary points.*

Step 1. Start with any point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Apply Algorithm 3.2 setting $x = x^k$. This algorithm terminates after finite number of iterations. Thus, we have the set $\bar{H}_m(x^k) \subset H(x, \lambda) \subset \underline{d}f(x)$ and an element $\bar{w}^k = (\bar{a}_k, \bar{v}^k)$ such that

$$\|\bar{w}^k\|^2 = \min\left\{\|w\|^2 \mid\ w \in \bar{H}_m(x^k)\right\}.$$

Moreover, either

$$\|\bar{w}^k\| \le \delta \tag{3.16}$$

or

$$f(x^k + \lambda g^k) - f(x^k) \le -c_1\lambda\|\bar{w}^k\| \tag{3.17}$$

for the search direction $g^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ holds.

Step 3. If $\|\bar{w}^k\| \le \delta$, then **stop**. Otherwise, go to Step 4.

Step 4. Compute $x^{k+1} = x^k + \alpha_k g^k$, where α_k is defined as follows

$$\alpha_k = \operatorname{argmax}\left\{\alpha \ge 0 :\ f(x^k + \alpha g^k) - f(x^k) \le -c_2\alpha\|\bar{w}^k\|\right\}. \tag{3.18}$$

Set $k \leftarrow k + 1$ and go to Step 2.

The following theorem shows that Algorithm 3.5 stops after finite number of iterations and it gives an upperbound for the number of iterations.

Theorem 3.6 *Assume that the function f is bounded from below:*

$$f_* = \inf\{f(x) \mid x \in \mathbb{R}^n\} > -\infty. \tag{3.19}$$

Then, Algorithm 3.5 terminates after finite number $M > 0$ of iterations. As a result, this algorithm generates a (λ, δ)-stationary point x^M, where

$$M \le M_0 \equiv \left\lfloor \frac{f(x^0) - f_*}{c_2\lambda\delta} \right\rfloor + 1.$$

Proof: Assume the statement in the theorem is not correct. Then, we have infinite sequence $\{x^k\}$ and non-(λ, δ)-stationary points x^k. This means that

$$\min\left\{\|w\| \mid\ w \in H(x^k, \lambda)\right\} > \delta \quad \forall k \in \mathbb{N}.$$

Therefore, Algorithm 3.2 will find descent directions by satisfying the inequality (3.17). Since $c_2 \in (0, c_1]$, it follows from (3.17) that $\alpha_k \geq \lambda$. Therefore, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &< -c_2\alpha_k\|w^k\| \\ &\leq -c_2\lambda\|w^k\|. \end{aligned}$$

Since $\|w^k\| > \delta$ for all $k \geq 0$, we get

$$f(x^{k+1}) - f(x^k) \leq -c_2\lambda\delta,$$

which implies

$$f(x^{k+1}) \leq f(x^0) - (k+1)c_2\lambda\delta$$

and, therefore, $f(x^k) \to -\infty$ as $k \to +\infty$ which contradicts (3.19). Obviously, in order to find the (λ, δ)-stationary point, the upper bound of iterations is M_0 △

Remark 3.7 Because of the fact that $c_2 \leq c_1$ and $\alpha_k \geq \lambda$, $\lambda > 0$ is a lower bound for α_k. This allows us to estimate α_k by using the following rule:
α_k is defined as the largest $\theta_l = 2^l\lambda \quad (l \in \mathbb{N})$, satisfying the inequality in Equation 3.18.

Now, an algorithm for solving Problem (3.1) will be described. The sequences $\{\lambda_k\}$, $\{\delta_k\}$ must be satisfied the conditions $\lambda_k \to +0$ and $\delta_k \to +0$ $(k \to \infty)$. The tolerances $\varepsilon_{opt} > 0$, $\delta_{opt} > 0$ must be given.

Algorithm 3.8 *The truncated codifferential method.*

Step 1. Start with any point $x^0 \in \mathbb{R}^n$, and set $k = 0$.

Step 2. If $\lambda_k \leq \varepsilon_{opt}$ and $\delta_k \leq \delta_{opt}$, then terminates.

Step 3. Apply Algorithm 3.5 setting initial point as x^k and the tolerances $\lambda = \lambda_k$ and $\delta = \delta_k$. This algorithm stops after a finitely many iterations. As a result, a (λ_k, δ_k)-stationary point x^{k+1} is generated.

Step 4. Set $k \leftarrow k + 1$ and continue from Step 2.

Consider the set $\mathcal{L}(x^0) = \left\{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\right\}$ for the point $x^0 \in \mathbb{R}^n$.

Theorem 3.9 *Assume that the function f is proper convex, the set $\mathcal{L}(x^0)$ is bounded for starting point x_0. Then, every accumulation point of the sequence $\{x^k\}$ generated by Algorithm 3.8 belongs to the set $X^0 = \{x \in \mathbb{R}^n \mid 0_n \in \partial f(x)\}$.*

Proof: Since the function f is proper convex and the set $\mathcal{L}(x^0)$ is bounded, $f_* > -\infty$. Therefore, conditions of Theorem 3.6 are satisfied, and Algorithm 3.5 generates a sequence of (λ_k, δ_k)-stationary points for all $k \geq 0$. More specifically, the point x^{k+1} is (λ_k, δ_k)-stationary, $k > 0$. Then, it follows from Definition 3.4 that

$$\min\left\{\|w\| \mid w \in H(x^{k+1}, \lambda_k)\right\} \leq \delta_k. \tag{3.20}$$

It is obvious that $x^k \in \mathcal{L}(x^0)$ for all $k \geq 0$. The boundedness of the set $\mathcal{L}(x^0)$ implies that the sequence $\{x^k\}$ has at least one accumulation point. Let x^* be an accumulation point and $x^{k_i} \to x^*$ as $i \to +\infty$. The inequality in (3.20) implies that

$$\min\left\{\|w\| \mid w \in H(x^{k_i}, \lambda_{k_i-1})\right\} \leq \delta_{k_i-1}.$$

Then, there exists $\bar{w} \in H(x^{k_i}, \lambda_{k_i-1})$ such that $\|\bar{w}\| \leq \delta_{k_i-1}$. Considering $\bar{w} = (\bar{a}, \bar{v})$ where $\bar{v} \in \partial f(y)$ for some $y \in B_{\lambda_{k_i-1}}(x^{k_i})$, we have $\|\bar{v}\| \leq \|\bar{w}\| \leq \delta_{k_i-1}$. Therefore,

$$\min\left\{\|v\| \mid \ v \in \partial f(x^{k_i} + B_{\lambda_{k_i-1}}(x^{k_i}))\right\} \leq \delta_{k_i-1},$$

where

$$\partial f(x^{k_i} + B_{\lambda_{k_i-1}}(x^{k_i})) = \bigcup \left\{\partial f(y) \mid \ y \in B_{\lambda_{k_i-1}}(x^{k_i})\right\}.$$

The upper semicontinuity of the subdifferential mapping $\partial f(x)$ implies that for any $\varepsilon > 0$ there exists $\eta > 0$ such that

$$\partial f(y) \subset \partial f(x^*) + B_\varepsilon(0_n) \tag{3.21}$$

for all $y \in B_\eta(x^*)$. Since $x^{k_i} \to x^*, \ \delta_{k_i}, \lambda_{k_i} \to +0 \ (i \to +\infty)$, there exists $i_0 > 0$ such that $\delta_{k_i} < \varepsilon$ and

$$B_{\lambda_{k_i-1}}(x^{k_i}) \subset B_\eta(x^*)$$

for all $i \geq i_0$. Then, it follows from (3.21) that

$$\min\{\|v\| \mid \ v \in \partial f(x^*)\} \leq 2\varepsilon.$$

Since $\varepsilon > 0$ has been chosen arbitrarily, we have $0 \in \partial f(x^*)$. △

CHAPTER 4

TRUNCATED CODIFFERENTIAL METHOD WITH MEMORY

In this chapter, a new method for solving unconstrained nonsmooth convex optimization problems will be introduced. The main difference between this method and the truncated codifferential method in Chapter 3 is that at each iteration of the algorithm proposed in this study one uses also codifferential computed at previous iterations. Because of that, we shall call this method as *Truncated codifferential method with memory*. Using codifferential from previous iterations will allow us to reduce the number of function and subgradient evaluations respectably when compared with the truncated codifferential method. The convergence of the proposed method will be proved. Results of numerical experiments using a set of test problems with not only nonsmooth convex but also nonsmooth nonconvex objective function will be reported in Chapter 6 by comparing the proposed algorithm with TCM and some other algorithms.

4.1 Introduction

In this chapter, similarly Chapter 3, the solution of unconstrained convex optimization problem is focused. The problem is as follows:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array} \tag{4.1}$$

where the objective function f is assumed to be proper convex.

Several numerical techniques to find the solution of Problem (4.1) have been developed in the literature. Subgradient method [83], different version of bundle methods [33, 34, 35, 36, 39, 42, 47, 49, 55, 59, 58, 64, 65, 67, 86] are among them. On the other hand, the number of the studies which use codifferential is just a few, because it is considered that either the entire codifferential or its subsets should be computed at any point. Whereas, these assumptions are too restrictive. Actually their calculations are not possible for many class of nonsmooth functions.

In this chapter, we introduce a new method for solving unconstrained nonsmooth convex optimization problems. The main difference between this method and the truncated codifferential method [12] is that at each iteration of the algorithm proposed in this study one uses also codifferential computed at previous iterations. Because of that, we call this method as *truncated codifferential method with memory*. This approach reduces the number of function and subgradient evaluations when comparing with Truncated codifferential method.

This chapter is structured as follows; Section 4.2 presents an algorithm for finding descant directions. In Section 4.3, the method will be proposed to find minimum of Problem (4.1). Numerical results are reported in Section 6.3.

4.2 Computation of a Descent Direction

When computing search directions, we will use a subset of the hypodifferential which is given in the following way. To find the descent direction these computed search directions are used and usability of them will be proved. For any $\lambda \in (0,1)$, $c > 1$, we assume x^i for $i = 1, 2, ..., k-1$, are given and define the following set:

$$\mathscr{H} = \text{cl co} \left\{ H(x,\lambda) \cup \widetilde{H}(x,\lambda) \right\}, \tag{4.2}$$

where

$$H(x,\lambda) = \left\{ w = (a,v) \in \mathbb{R} \times \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists y \in B_\lambda(x), \\ v \in \partial f(y), \\ a = f(y) - f(x) - \langle v, y - x \rangle \end{array} \right\} \tag{4.3}$$

and

$$\widetilde{H}(x,\lambda) = \left\{ w = (a,v) \in \mathbb{R} \times \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists\, x^i, i = 1, ..., k-1 \\ \text{such that } \left\| x - x^i \right\| \le c\lambda, \\ v \in \partial f(y), \text{ where } y \in B_\lambda(x^i), \\ a = f(y) - f(x) - \langle v, y - x \rangle \end{array} \right\}. \tag{4.4}$$

Obviously, $a \le 0$ for all $w = (a, v) \in \mathscr{H}(x,\lambda)$ because of definition of the subdifferential. Since y can take the value x in the set (4.3), a attains the value 0, so

$$\max_{w=(a,v) \in \mathscr{H}(x,\lambda)} a = 0. \tag{4.5}$$

If $\bar{B}_{(c+1)\lambda}(x) \subset \text{int} U$ for all $\lambda \in (0,1)$ where $U \subset \mathbb{R}^n$ is a closed convex set, then from the definition of both the hypodifferential and the set $\mathscr{H}(x,\lambda)$, the following inclusion holds:

$$\mathscr{H}(x,\lambda) \subset \underline{d} f(x) \;\; \forall\, \lambda \in (0,1).$$

We call the sets $\mathscr{H}(x,\lambda)$ as the truncated codifferential with memory of the function f at the point x.

Proposition 4.1 *Let us assume that $0_{n+1} \notin \mathscr{H}(x,\lambda)$ for a given $\lambda \in (0,1)$ and $c \ge 1$, $(c+1)\lambda \in (0,1)$ and*

$$\|w^0\| = \min \{ \|w\| \mid w \in \mathscr{H}(x,\lambda) \} > 0, \; \textit{with} \; w^0 = (a_0, v^0). \tag{4.6}$$

Then, $v^0 \ne 0_n$ and

$$f(x + \lambda g^0) - f(x) \le -\lambda \|w^0\|, \tag{4.7}$$

where $g^0 = -\|w^0\|^{-1} v^0$.

Proof: Since w^0 is a solution of 4.6,

$$\langle w^0, w \rangle \ge \langle w^0, w^0 \rangle \quad \forall w = (a,v) \in \mathscr{H}(x,\lambda)$$

or

$$a_0 a + \langle v^0, v \rangle \ge \|w^0\|^2. \tag{4.8}$$

First, $v^0 \ne 0_n$ should be proved. Assume $v^0 = 0_n$. Since $w^0 \ne 0_{n+1}$ we get that $a_0 < 0$ (i.e., $a \ne 0$). Then, it follows from (4.8) that $a_0 a \ge a_0^2$ or $a \le a_0 < 0$. In other words, $a < 0$ for all $w = (a, v) \in$

$\mathscr{H}(x, \lambda)$ which contradicts (4.5). Now we will prove (4.7). Obviously $x + \lambda g^0 \in B_\lambda(x)$ and that implies $a = f(x+\lambda g^0) - f(x) - \lambda\langle v, g^0\rangle$, with $v \in \partial f(x+\lambda g^0)$ because of the definition of the set $H(x, \lambda)$. Thus, $w = (a, v) \in \mathscr{H}(x, \lambda)$ with above given a and v. Replacing a with $f(x + \lambda g^0) - f(x) - \lambda\langle v, g^0\rangle$ in (4.8), we get

$$a_0(f(x + \lambda g^0) - f(x) - \lambda\langle v, g^0\rangle) + \langle v, v^0\rangle \geq \|w^0\|^2. \tag{4.9}$$

Dividing (4.9) by $-\|w^0\|$ and multiplying (4.9) by λ, we obtain the following inequality

$$-\frac{\lambda a_0}{\|w^0\|}(f(x + \lambda g^0) - f(x) - \lambda\langle v, g^0\rangle) + \lambda\langle v, g^0\rangle \leq -\lambda\|w^0\|. \tag{4.10}$$

Obviously since $\lambda \in (0, 1)$ and $\|w^0\|^{-1}a_0 \in (-1, 0)$, $-\dfrac{\lambda a_0}{\|w^0\|} \in (0, 1)$ and $a = f(x+\lambda g^0) - f(x) - \lambda\langle v, g^0\rangle \leq 0$, so (4.10) gives the following inequality

$$\begin{aligned} &f(x + \lambda g^0) - f(x) - \lambda\langle v, g^0\rangle + \lambda\langle v, g^0\rangle \leq -\lambda\|w^0\|, \\ &f(x + \lambda g^0) - f(x) \leq -\lambda\|w^0\|. \end{aligned}$$

$\triangle$

According to Proposition 4.1, the set $\mathscr{H}(x, \lambda)$ can be used to find descent directions of a function f. Moreover, this can be done for any $\lambda \in (0, 1)$. Unfortunately, it is generally not possible to find descent direction by using Proposition 3.1 since the entire set $\mathscr{H}(x, \lambda)$ must be used. Actually, the usage of the entire set $\mathscr{H}(x, \lambda)$ is not always possible. However, Proposition 4.1 helps us how an algorithm for finding descent directions can be developed. In order to over come this difficulty, the following algorithm is developed using only a few elements from $\mathscr{H}(x, \lambda)$ to compute descent directions.

Assume that from previous iterations, we have some information about the point x^i for $i = 1, 2, ..., k-1$, namely subgradients of the function f at the point x^i for $i = 1, 2, ..., k-1$, and the points related that subgradients. Also assume that the number of that subgradients is finite and it is denoted m_i for $i = 1, 2, ..., k-1$. Let the subgradients of x^i for $i = 1, 2, ..., k-1$, be denoted v^i_j and related points be denoted y^i_j for $j = 1, 2, ..., m_i$ and $i = 1, 2, ..., k-1$. Consider the following set

$$\widetilde{H}(x) = \left\{ w = (a, v) \in \mathbb{R} \times \mathbb{R}^n \;\middle|\; \begin{array}{l} \exists\ x^i, i = 1, ..., k-1 \\ \text{such that } \left\|x - x^i\right\| \leq c\lambda \\ v = v^i_j \text{ and } a = f(y^i_j) - f(x) - \langle v^i_j, y^i_j - x\rangle, \\ \text{ for } j = 1, 2, ..., m_i \end{array} \right\}$$

Now, we can give the algorithm which compute the descent direction. For the given numbers λ, $c \in (0, 1)$ and a small enough number $\delta > 0$, the following algorithm can be used to find descent directions.

Algorithm 4.2 *Computation of descent directions at x.*

Step 1. If $\widetilde{H}(x) \neq \emptyset$, then set $k = \left|\widetilde{H}(x)\right|$, $\bar{H}_k(x) = \operatorname{co}\{\widetilde{H}(x)\}$ and go to Step 2. Otherwise, select any $g^1 \in S_1$, and compute $v^1 \in \partial f(x+\lambda g^1)$ and $a_1 = f(x+\lambda g^1) - f(x) - \lambda\langle v^1, g^1\rangle$. Set $\bar{H}_1(x) = \{w^1 = (a_1, v^1)\}$ and $k = 1$.

Step 2. Compute $\bar{w}^k = (\bar{a}_k, \bar{v}^k) \in \mathbb{R} \times \mathbb{R}^n$ as a solution to the following problem:

$$\min \quad \|w\|^2 \quad \text{subject to } \; w \in \bar{H}_k(x). \tag{4.11}$$

Step 3. If

$$\|\bar{w}^k\| \leq \delta, \tag{4.12}$$

then **stop**. Otherwise, compute $\bar{g}^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ and go to Step 4.

Step 4. If

$$f(x+\lambda\bar{g}^k) - f(x) \leq -c\lambda\|\bar{w}^k\|, \tag{4.13}$$

then **stop**. Otherwise, set $g^{k+1} = \bar{g}^k$ and go to Step 5.

Step 5. Compute $v^{k+1} \in \partial f(x+\lambda g^{k+1})$ and $a_{k+1} = f(x+\lambda g^{k+1}) - f(x) - \lambda\langle v^{k+1}, g^{k+1}\rangle$. Construct the set $\bar{H}_{k+1}(x) = \operatorname{co}\{\bar{H}_k(x) \bigcup \{w^{k+1} = (a_{k+1}, v^{k+1})\}\}$, set $k \leftarrow k+1$ and go to Step 2.

Some explanations on Algorithm 4.2 follow. In Step 1, if we have some hypogradients from known information, we start to find descent direction by using them. Otherwise, we select any direction $g^1 \in S_1$ and compute the hypogradient in this direction and star to find descent direction by using it. The closest point to the origin in the set of all computed codifferential is computed in Step 2. This problem is a quadratic optimization problem. In the literature there are several algorithms [32, 48, 73, 74, 87] to solve this problem. In the implementation the algorithm from [87] is used. If the norm of the closest point is less than a given tolerance $\delta > 0$, then the point x is an approximate stationary point; otherwise, a new search direction is computed in Step 3. In Step 4, we check whether it is descent direction satisfying the inequality (4.13) or not. If it is descent direction, then the algorithm stops. Otherwise, we compute a new hypogradient in the direction g^{k+1} in Step 5.

Proposition 4.3 *Let us assume that f is proper convex function, given a number $\lambda \in (0,1)$ and there exists a value $K \in (0,\infty)$ such that*

$$\max\left\{\|w\| \mid w \in \underline{d}f(x)\right\} \leq K.$$

If $c \in (0,1)$ and $\delta \in (0,K)$, then Algorithm 4.2 terminates after at most m steps, where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 1, \quad K_1 = 1 - 2[(1-c)(2K)^{-1}\delta]^2.$$

Proof: First, we will show that if both stopping criteria (4.12) and (4.13) are not satisfied, then a new hypogradient w^{k+1} allows us to improve to the set $\bar{H}_k(x)$. Let us assume the contrary, that is $w^{k+1} \in \bar{H}_k(x)$. Since both stopping criteria are not satisfied, we have

$$\|\bar{w}^k\| > \delta$$

and

$$f(x+\lambda g^{k+1}) - f(x) > -c\lambda\|\bar{w}^k\|. \tag{4.14}$$

The definition of the hypogradient $w^{k+1} = (a_{k+1}, v^{k+1})$ implies that

$$f(x+\lambda g^{k+1}) - f(x) = a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle, \tag{4.15}$$

where $g^{k+1} = \bar{g}^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$. Combining (4.14) and (4.15), we have

$$-c\lambda\|\bar{w}^k\| < a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle.$$

Putting $g^{k+1} = -\|\bar{w}^k\|^{-1}\bar{v}^k$ and multiplying by $\dfrac{\|\bar{w}^k\|}{\lambda}$, we get

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda}a_{k+1} < c\|\bar{w}^k\|^2. \tag{4.16}$$

Since $\bar{w}^k = \operatorname{argmin}\{\|w\|^2 : w \in \bar{H}_k(x)\}$, the necessary condition for a minimum implies that

$$\langle \bar{w}^k, w\rangle \geq \|\bar{w}^k\|^2$$

for all $w \in \bar{H}_k(x)$. By the assumption, $w^{k+1} \in \bar{H}_k(x)$ holds. By replacing w with w^{k+1}, we get

$$a_{k+1}\bar{a}_k + \langle \bar{v}^k, v^{k+1}\rangle \geq \|\bar{w}^k\|^2. \tag{4.17}$$

We notice that $a_{k+1} \leq 0$ and $\bar{a}_k \geq -\|\bar{w}^k\|$. Then, we have $\bar{a}_k a_{k+1} \leq -\|\bar{w}^k\| a_{k+1}$. Combining this with (4.17), we obtain

$$\langle v^{k+1}, \bar{v}^k\rangle - \|\bar{w}^k\| a_{k+1} \geq \|\bar{w}^k\|^2.$$

Finally, taking into account that $\lambda \in (0,1)$, we have

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda} a_{k+1} \geq \|\bar{w}^k\|^2,$$

which contradicts (4.16). Thus, if both stopping criteria are not satisfied, then the new hypogradient w^{k+1} does not belongs to the set $\bar{H}_{k+1}(x)$. In light of this fact $\bar{H}_{k+1}(x)$ approximates to the set $H(x,\lambda)$.

Since at a point x for a given $\lambda \in (0,1)$ it holds

$$\bar{H}_k(x) \subset \widetilde{H}(x,\lambda) \subset \underline{d} f(x)$$

for any $k = 1, 2, \ldots$, it follows that

$$\max\left\{\|w\| \mid w \in \bar{H}_k(x)\right\} \leq K \quad \forall k \in \mathbb{N}. \tag{4.18}$$

Obviously, $\|\bar{w}^{k+1}\|^2 \leq \|tw^{k+1} + (1-t)\bar{w}^k\|^2$ for all $t \in [0,1]$, which means

$$\|\bar{w}^{k+1}\|^2 \leq \|\bar{w}^k\|^2 + 2t\langle \bar{w}^k, w^{k+1} - \bar{w}^k\rangle + t^2\|w^{k+1} - \bar{w}^k\|^2.$$

Inequality (4.18) implies that

$$\|w^{k+1} - \bar{w}^k\| \leq 2K.$$

It follows from (4.16) that

$$\begin{aligned}
\langle \bar{w}^k, w^{k+1}\rangle &= \bar{a}_k a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \\
&\leq -\frac{\|\bar{w}^k\|}{\lambda} a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \\
&\leq -c\|\bar{w}^k\|^2.
\end{aligned}$$

Then, we have

$$\|\bar{w}^{k+1}\|^2 \leq \|\bar{w}^k\|^2 - 2t(1-c)\|\bar{w}^k\|^2 + 4t^2K^2.$$

Let $t = (1-c)(2K)^{-2}\|\bar{w}^k\|^2$. It is clear that $t \in (0,1)$ and, therefore,

$$\|\bar{w}^{k+1}\|^2 \leq \left\{1 - [(1-c)(2K)^{-1}\|\bar{w}^k\|]^2\right\}\|\bar{w}^k\|^2. \tag{4.19}$$

Since $\|\bar{w}^k\| > \delta$ for all $k = 1, \ldots, m-1$, it follows from (4.19) that

$$\|\bar{w}^{k+1}\|^2 \leq \{1 - [(1-c_1)(2K)^{-1}\delta]^2\}\|\bar{w}^k\|^2.$$

Let $K_1 = 1 - [(1-c_1)(2K)^{-1}\delta]^2$. Then, $K_1 \in (0,1)$ and we have

$$\|\bar{w}^m\|^2 \leq K_1\|\bar{w}^{m-1}\|^2 \leq \ldots \leq K_1^{m-1}\|\bar{w}^1\|^2 \leq K_1^{m-1}K^2.$$

Thus, the inequality $\|\bar{w}^m\| \leq \delta$ is satisfied if $K_1^{m-1}K^2 \leq \delta^2$. This inequality must happen after at most m steps where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 1.$$

△

Definition 4.4 *A point $x \in \mathbb{R}^n$ is called a (λ, δ)-stationary point of the function f if*

$$\min_{w \in \mathscr{H}(x,\lambda)} \|w\| \leq \delta.$$

It can be easily observed that Algorithm 4.2 for a given point x either finds a descent direction or determines the point x as a (λ, δ)-stationary point for the convex function f.

4.3 A Codifferential Method

In this section, the truncated codifferential method to find the solution of problem (3.1) is introduced. First of all, we should find stationary points with some tolerance. According to this purpose, the following algorithm was designed to find (λ, δ)-stationary points for given numbers $\lambda \in (0, 1)$, $c_1 \in (0, 1)$, $c_2 \in (0, c_1]$ and the tolerance $\delta > 0$.

Algorithm 4.5 *Finding (λ, δ)-stationary points.*

Step 1. Start with any point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Apply Algorithm 4.2 setting $x = x^k$. This algorithm terminates after finite number of iterations. Thus, we have the set $\bar{H}_m(x^k) \subset \widetilde{H}(x, \lambda) \subset \underline{d}f(x)$ and an element $\bar{w}^k = (\bar{a}_k, \bar{v}^k)$ such that

$$\|\bar{w}^k\|^2 = \min\left\{\|w\|^2 \mid \ w \in \bar{H}_m(x^k)\right\}.$$

Furthermore, either

$$\|\bar{w}^k\| \leq \delta \tag{4.20}$$

or

$$f(x^k + \lambda g^k) - f(x^k) \leq -c_1\lambda\|\bar{w}^k\|. \tag{4.21}$$

for the search direction $g^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ holds *Step 3.* If $\|\bar{w}^k\| \leq \delta$, then **stop**. Otherwise, go to Step 4.

Step 4. Compute $x^{k+1} = x^k + \alpha_k g^k$, where α_k is defined as follows

$$\alpha_k = \operatorname{argmax} \left\{\alpha \geq 0 \mid \ f(x^k + \alpha g^k) - f(x^k) \leq -c_2\alpha\|\bar{w}^k\|\right\}. \tag{4.22}$$

Set $k \leftarrow k + 1$ and go to Step 2.

Theorem 4.6 *Let us assume that the function f is bounded below, i.e.*

$$f_* = \inf\{f(x) \mid x \in \mathbb{R}^n\} > -\infty. \tag{4.23}$$

Then, Algorithm 4.5 terminates after a finite number $M > 0$ of iterations and generates a (λ, δ)-stationary point x^M, where

$$M \leq M_0 \equiv \left\lfloor \frac{f(x^0) - f_*}{c_2\lambda\delta} \right\rfloor + 1.$$

Proof: Assume the statement in the theorem is not correct. Then, we have infinite sequence $\{x^k\}$ and non-(λ, δ)-stationary points x^k. This means that

$$\|\bar{w}^k\| = \min\left\{\|w\| \mid w \in \mathscr{H}(x^k, \lambda)\right\} > \delta \ (k = 1, 2, \ldots).$$

Therefore, Algorithm 4.2 will find descent directions by satisfying the inequality (4.21). Since $c_2 \in (0, c_1]$, it follows from (4.21) that $\alpha_k \geq \lambda$. Thus, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &< -c_2\alpha_k\|\bar{w}^k\| \\ &\leq -c_2\lambda\|\bar{w}^k\|. \end{aligned}$$

Since $\|\bar{w}^k\| > \delta$ for all $k \geq 0$, we get

$$f(x^{k+1}) - f(x^k) \leq -c_2\lambda\delta,$$

which implies

$$f(x^{k+1}) \leq f(x^0) - (k+1)c_2\lambda\delta$$

and, therefore, $f(x^k) \to -\infty$ as $k \to +\infty$, which contradicts (4.23). Obviously, in order to find the (λ, δ)-stationary point, the upper bound of iterations is M_0 $\triangle$

In the calculation of α_k the following idea is used. Because of the fact that $c_2 \leq c_1$ and $\alpha_k \geq \lambda$, $\lambda > 0$ is a lower bound for α_k. This allows us to estimate α_k by using the following rule:
α_k is defined as the largest $\theta_l = 2^l\lambda \quad (l \in \mathbb{N})$, satisfying the inequality in Equation 4.22.

Now, an algorithm for solving Problem (4.1) will be described. The sequences $\{\lambda_k\}$, $\{\delta_k\}$ must be satisfied the conditions $\lambda_k \to +0$ and $\delta_k \to +0$ $(k \to \infty)$. The tolerances $\varepsilon_{opt} > 0$, $\delta_{opt} > 0$ must be given.

Algorithm 4.7 *The truncated codifferential method.*

Step 1. Start with any point $x^0 \in \mathbb{R}^n$, and set $k = 0$.

Step 2. If $\lambda_k \leq \varepsilon_{opt}$ and $\delta_k \leq \delta_{opt}$, then stop.

Step 3. Apply Algorithm 4.5 setting initial point as x^k and the tolerances $\lambda = \lambda_k$ and $\delta = \delta_k$. This algorithm stops after a finitely many iterations. As a result, a (λ_k, δ_k)-stationary point x^{k+1} is generated.

Step 4. Set $k \leftarrow k + 1$ and go to Step 2.

For the point $x^0 \in \mathbb{R}^n$, we consider the set $\mathcal{L}(x^0) = \left\{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\right\}$.

Theorem 4.8 *Assume that f is a proper convex function and the set $\mathcal{L}(x^0)$ is bounded. Then, every accumulation point of the sequence $\{x^k\}$ generated by Algorithm 4.7 belongs to the set $X^0 = \{x \in \mathbb{R}^n \mid 0_n \in \partial f(x)\}$.*

Proof: Since the function f is proper convex and the set $\mathcal{L}(x^0)$ is bounded, $f_* > -\infty$. Therefore, the conditions of Theorem 4.6 are satisfied, and Algorithm 4.5 generates a sequence of (λ_k, δ_k)-stationary points for all $k \geq 0$. More specifically, the point x^{k+1} is (λ_k, δ_k)-stationary, $k > 0$. Then, it follows from Definition 4.4 that

$$\min\left\{\|w\| \mid w \in \mathscr{H}(x^{k+1}, \lambda_k)\right\} \leq \delta_k. \tag{4.24}$$

It is obvious that $x^k \in \mathcal{L}(x^0)$ for all $k \geq 0$. The boundedness of the set $\mathcal{L}(x^0)$ implies that the sequence $\{x^k\}$ has at least one accumulation point. Let x^* be an accumulation point and $x^{k_i} \to x^*$ as $i \to +\infty$. The inequality in (4.24) implies that

$$\min\left\{\|w\| \mid w \in \mathscr{H}(x^{k_i}, \lambda_{k_i-1})\right\} \leq \delta_{k_i-1}.$$

Then, there exists a point $\bar{w} \in \mathscr{H}(x^{k_i}, \lambda_{k_i-1})$ such that $\|\bar{w}\| \leq \delta_{k_i-1}$. Considering $\bar{w} = (\bar{a}, \bar{v})$ where $\bar{v} \in \partial f(y)$ for some $y \in B_{(c+1)\lambda_{k_i-1}}(x^{k_i})$ for $c \geq 1$ given in definition of $\mathscr{H}(x, \lambda)$, we have $\|\bar{v}\| \leq \|\bar{w}\| \leq \delta_{k_i-1}$. Therefore,

$$\min\left\{\|v\| \mid v \in \partial f(B_{(c+1)\lambda_{k_i-1}}(x^{k_i}))\right\} \leq \delta_{k_i-1}.$$

Here

$$\partial f(B_{(c+1)\lambda_{k_i-1}}(x^{k_i})) = \bigcup \left\{\partial f(y) \mid y \in B_{(c+1)\lambda_{k_i-1}}(x^{k_i})\right\}.$$

The upper semicontinuity of the subdifferential mapping $\partial f(x)$ implies that for any $\varepsilon > 0$ there exists a number $\eta > 0$ such that

$$\partial f(y) \subset \partial f(x^*) + B_\varepsilon(0_n) \tag{4.25}$$

for all $y \in B_\eta(x^*)$. Since $x^{k_i} \to x^*,\ \delta_{k_i}, \lambda_{k_i} \to +0$ as $i \to +\infty$, there exists an $i_0 > 0$ such that

$$\delta_{k_i} < \varepsilon \quad \text{and} \quad B_{(c+1)\lambda_{k_i-1}}(x^{k_i}) \subset B_\eta(x^*) \text{ for all } i \geq i_0.$$

Then, it follows from (4.25) that

$$\min\{\|v\| \mid\ v \in \partial f(x^*)\} \leq 2\varepsilon.$$

Since $\varepsilon > 0$ has been chosen arbitrarily, we have $0 \in \partial f(x^*)$. △

CHAPTER 5

AGGREGATE CODIFFERENTIAL METHOD

In this chapter, another method for nonsmooth convex optimization problem will be developed via codifferential concept. Similar to the truncated codifferential method (TCM) given in Chapter 3, we shall use a few elements from the codifferential. The difference between the method which will be mentioned in this chapter and TCM is that a fixed number elements of the codifferential will be used to compute descent direction at each iteration. The convergence of the proposed minimization algorithm will be proved. Results of numerical experiments using a set of test problems with not only nonsmooth convex but also nonsmooth nonconvex objective function will be reported in Chapter 6 by comparing the proposed method with TCM, the truncated codifferential method with memory (TCMWM) and some other well known methods.

5.1 Introduction

In this chapter, we develop an algorithm for solving the following unconstrained nonsmooth optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbb{R}^n \end{array} \tag{5.1}$$

where the objective function f is assumed to be proper convex.

There are a number of methods in nonsmooth optimization for solving Problem (5.1). We mention a few here such as the subgradient method [83], different versions of the bundle method [33, 36, 49, 55, 58, 64, 67, 86], the variable metric method [59] and the discrete gradient method [14].

The proposed method is based on the concept of codifferential, which was introduced in [27]. The codifferential mapping for most of the important classes of nonsmooth functions is Hausdorff continuous. Although it has good differential properties, only very few numerical methods were developed based on the codifferential [26, 27, 91]. These algorithms [26, 27, 91] require either the computation of whole codifferential or its subsets at any point. However, this assumption is too restrictive for many nonsmooth optimization problems.

In order to overcome this, the truncated codifferential method (TCM) was developed in [12], where only one element of the codifferential is computed at any point. Numerical experiments show that the TCM is a robust and efficient method for solving Problem (5.1). However, the size of the bundle used to find search directions is not fixed which may lead to a large scale quadratic programming problem to be solved at each iteration. It is therefore desirable to develop a modification of the TCM, where the size of this bundle is fixed. In this paper, we develop one such modification. The proposed algorithm uses aggregate codifferentials to preserve the efficiency and robustness of the TCM. We study the convergence of the algorithm and demonstrate its efficiency on well-known nonsmooth optimization test problems by comparing with the subgradient, the truncated codifferential and the proximal bundle methods.

This chapter is structured as follows: how the codifferential is used in order to find descent direction is explained in Subsection 5.2. The aggregate codifferential method is given and its convergency is examined in Subsection 5.3. The results is presented in Subsection 6.3.

5.2 Computation of a Descent Direction

In the previous section we mentioned difficulty of computation of whole hypodifferential $\underline{d}f(x)$. In order to overcome this difficulty truncated codifferential of the function f at the point x is defined as a subset of the hypodifferential in following definition;

Definition 5.1 *Let $\lambda \in (0,1)$, then*

$$H(x,\lambda) = cl\ co \left\{ w = (a,v) \in \mathbb{R} \times \mathbb{R}^n : \begin{array}{l} \exists y \in B(x,\lambda), \\ v \in \partial f(y), \\ a = f(y) - f(x) - \langle v, y - x \rangle \end{array} \right\}. \tag{5.2}$$

is called truncated codifferential *of the function f at the point x.*

Clearly, $H(x,\lambda) \subset \underline{d}f(x)\ \ \forall\ \lambda \in (0,1)$ for any convex function f because of the definition of both the hypodifferential and the set $H(x,\lambda)$. On the other hand, it can be easily observed that $a \leq 0$ for all $w = (a,v) \in H(x,\lambda)$ and $a = 0$ at the point x, so

$$\max_{w=(a,v)\in H(x,\lambda)} a = 0. \tag{5.3}$$

We have proved that $H(x,\lambda)$ can be used to find a descent direction in Proposition 3.1. The following algorithm gives the descent direction by using l codifferentials in each quadratic subproblem. When compared with Algorithm 3.2, although the size of quadratic Subproblem 3.9 can freely increase in Algorithm 3.2, the size of quadratic Subproblem 5.4 will be always less and equal l in the following algorithm.

Algorithm 5.2 *Computation of descent directions for given number l.*

Step 1. Choose any $g^1 \in S_1$, compute $w^1 = (a_1, v^1)$ as $v^1 \in \partial f(x+\lambda g^1)$, $a_1 = f(x+\lambda g^1) - f(x) - \lambda\langle v^1, g^1\rangle$ and set $k = 1$.

Step 2. If $k \leq l$, set $\bar{H}_k(x) = \text{co}\,\{w^1, ..., w^k\}$. Otherwise, set
$\bar{H}_k(x) = \text{co}\,\{\bar{w}^{k-l+1}, w^{k-l+2}, ..., w^{k-1}, w^k\}$.

Step 3. Compute $\bar{w}^k = (\bar{a}_k, \bar{v}^k) \in \mathbb{R} \times \mathbb{R}^n$ solving the quadratic subproblem:

$$\min \ \ \|w\|^2 \quad \text{subject to } w \in \bar{H}_k(x). \tag{5.4}$$

Step 4. If

$$\|\bar{w}^k\| \leq \delta, \tag{5.5}$$

then **stop**. Otherwise, compute $\bar{g}^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ and go to Step 5.

Step 5. If

$$f(x + \lambda\bar{g}^k) - f(x) \leq -c\lambda\|\bar{w}^k\|, \tag{5.6}$$

then **stop**. Otherwise, set $g^{k+1} = \bar{g}^k$ and go to Step 6.

Step 6. Compute $w^{k+1} = (a_{k+1}, v^{k+1})$, $v^{k+1} \in \partial f(x+\lambda g^{k+1})$ and $a_{k+1} = f(x+\lambda g^{k+1}) - f(x) - \lambda\langle v^{k+1}, g^{k+1}\rangle$. Set $k \leftarrow k + 1$ and go to Step 2.

In the following we give some explanation on Algorithm 5.2. In Step 1, we select any direction $g^1 \in S_1$ and compute the hypogradient in this direction. In Step 2, for a given number l, the subset of truncated codifferential is computed. In Step 3, the quadratic programming subproblem (5.4) is solved. It is used to find the closest point of the convex hull of $\bar{H}_k(x)$ to the origin. If the smallest distance is less than a tolerance $\delta > 0$, then the point x is an approximate stationary point; otherwise, a new search direction is computed in Step 4. In Step 5, it is checked whether the search direction satisfies the inequality (5.6). If yes, then the algorithm terminates. Otherwise, a new hypogradient is computed in Step 6 in the direction g^{k+1} to improve approximation of the truncated codifferential.

The following lemma proves $\bar{H}_k$ is a subset of $\underline{d}f(x)$ for all $k = 1, 2, ...$, $\bar{H}_k$ can be used to compute descent directions.

Lemma 5.3 *The set $\bar{H}_k$, which is generated by Algorithm 5.2 is a subset of $\underline{d}f(x)$ $(k \in \mathbb{N})$.*

Proof: It is obvious that $\bar{H}_k \subset \underline{d}f(x)$ for $k \leq l$, since $w^i \in \underline{d}f(x)$ for $i = 1, ..., k$. Let $k = l + 1$. Thus, $H_{l+1} = \text{co}\{\bar{w}^2, w^3, ..., w^l, w^{l+1}\}$.

Since $\bar{w}^1$ is solution of Problem (5.4) for H_l and $H_l \subset \underline{d}f(x)$, $\bar{w}^1 \in \underline{d}f(x)$, so that $H_{l+1} \subset \underline{d}f(x)$. Inductively, we can conclude $\bar{H}_k \subset \underline{d}f(x)$. △

In the following proposition we show that Algorithm 5.2 is finite convergent.

Proposition 5.4 *Assume that f is proper convex function, given a number $\lambda \in (0, 1)$ and there exists a value $K \in (0, \infty)$ such that*

$$\max\left\{\|w\| \mid w \in \underline{d}f(x)\right\} \leq K.$$

If $c \in (0, 1)$ and $\delta \in (0, K)$, then Algorithm 5.2 terminates after at most m steps, where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 2, \quad K_1 = 1 - [(1 - c)(2K)^{-1}\delta]^2.$$

Proof: First, we will show that if neither of the stopping criteria (5.5) or (5.6) are satisfied, then a new hypogradient w^{k+1} computed in Step 6 does not belong to the set $\bar{H}_k(x)$. Let us assume the contrary, that is, $w^{k+1} \in \bar{H}_k(x)$. In this case, $\|\bar{w}^k\| > \delta$ and

$$f(x + \lambda g^{k+1}) - f(x) > -c\lambda\|\bar{w}^k\|.$$

The definition of the hypogradient $w^{k+1} = (a_{k+1}, v^{k+1})$ implies that

$$f(x + \lambda g^{k+1}) - f(x) = a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle,$$

and we have

$$-c\lambda\|\bar{w}^k\| < a_{k+1} + \lambda\langle v^{k+1}, g^{k+1}\rangle.$$

Putting $g^{k+1} = -\|\bar{w}^k\|^{-1}\bar{v}^k$, we get

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda}a_{k+1} < c\|\bar{w}^k\|^2. \tag{5.7}$$

On the other hand, since $\bar{w}^k$ is the solution of Problem (5.4),

$$\langle \bar{w}^k, w\rangle \geq \|\bar{w}^k\|^2$$

for all $w \in \bar{H}_k(x)$. Since, by assumption, $w^{k+1} \in \bar{H}_k(x)$, we get

$$\bar{a}_k a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \geq \|\bar{w}^k\|^2. \tag{5.8}$$

We notice that $a_{k+1} \leq 0$ and since $\|\bar{w}^k\| = -a_k + \|v^k\|$, we obtain $\bar{a}_k \geq -\|\bar{w}^k\|$. Then, we have $\bar{a}_k a_{k+1} \leq -\|\bar{w}^k\|a_{k+1}$. Combining this with (5.8), we get

$$\langle v^{k+1}, \bar{v}^k\rangle - \|\bar{w}^k\|a_{k+1} \geq \|\bar{w}^k\|^2.$$

Finally, for any $\lambda \in (0,1)$, we get

$$\langle v^{k+1}, \bar{v}^k\rangle - \frac{\|\bar{w}^k\|}{\lambda}a_{k+1} \geq \|\bar{w}^k\|^2,$$

which contradicts (5.7). Thus, if both (5.5) and (5.6) do not hold then the new hypogradient w^{k+1} allows one to improve an other subset of $H(x,\lambda)$.

Because of the definition of $\bar{H}_k$ and computation of $\bar{w}^{k+1}$, $\|\bar{w}^{k+1}\|^2 \leq \|tw^{k+1} + (1-t)w^p\|^2$ for $\forall t \in [0,1]$ and $p = 1, ..., k$. That clearly implies $\|\bar{w}^{k+1}\|^2 \leq \|tw^{k+1} + (1-t)\bar{w}^k\|^2$ for all $t \in [0,1]$, which means

$$\|\bar{w}^{k+1}\|^2 \leq \|\bar{w}^k\|^2 + 2t\langle \bar{w}^k, w^{k+1} - \bar{w}^k\rangle + t^2\|w^{k+1} - \bar{w}^k\|^2.$$

By Lemma 5.3, we have

$$\|w^{k+1} - \bar{w}^k\| \leq 2K.$$

It follows from (5.7) that

$$\begin{aligned}
\langle \bar{w}^k, w^{k+1}\rangle &= \bar{a}_k a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \\
&\leq -\frac{\|\bar{w}^k\|}{\lambda}a_{k+1} + \langle \bar{v}^k, v^{k+1}\rangle \\
&< c\|\bar{w}^k\|^2.
\end{aligned}$$

Then, we have

$$\|\bar{w}^{k+1}\|^2 < \|\bar{w}^k\|^2 - 2t(1-c)\|\bar{w}^k\|^2 + 4t^2K^2.$$

Let $t_0 = (1-c)(2K)^{-2}\|\bar{w}^k\|^2$. It is clear that $t_0 \in (0,1)$ and, therefore,

$$\|\bar{w}^{k+1}\|^2 < \left\{1 - [(1-c)(2K)^{-1}\|\bar{w}^k\|]^2\right\}\|\bar{w}^k\|^2. \tag{5.9}$$

Since $\|\bar{w}^k\| > \delta$ for all $k = 1, \ldots, m-1$, it follows from (5.9) that

$$\|\bar{w}^{k+1}\|^2 < \{1 - [(1-c)(2K)^{-1}\delta]^2\}\|\bar{w}^k\|^2.$$

Let $K_1 = 1 - [(1-c)(2K)^{-1}\delta]^2$. Then, $K_1 \in (0,1)$ and we have

$$\|\bar{w}^m\|^2 < K_1\|\bar{w}^{m-1}\|^2 < \ldots < K_1^{m-1}\|\bar{w}^1\|^2 < K_1^{m-1}K^2.$$

Thus, the inequality $\|\overline{w}\| \leq \delta$ is satisfied if $K_1^{m-1}K^2 \leq \delta^2$. This inequality must take place after at most m steps, where

$$m \leq 2\log_2(\delta/K)/\log_2 K_1 + 1.$$

△

5.3 An Aggregate Codifferential Method

In this section, we will describe the algorithm for solving Problem (5.1) using the codifferential. First of all, the algorithm for finding a (λ, δ)-stationary point, which is defined in Definition 3.4, are given. Finally, an algorithm for solving problem (5.1) is described.

Obviously it can be observed that at a given point x after finitely many steps Algorithm 5.2 either finds a direction of sufficient decrease or determines that the point x is a (λ, δ)-stationary point of the convex function f. The following algorithm gives us a (λ, δ)-stationary point, when Algorithm 5.2 finds a descent direction.

Algorithm 5.5 *Computation of* (λ, δ)*-stationary points.*

Step 1. Start with any point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Apply Algorithm 5.2 to compute the descent direction at $x = x^k$ for given $\delta > 0$ and $c = c_1 \in (0, 1)$. This algorithm terminates after finite many steps $m_k > 0$. As a result, we get the set $\bar{H}_{m_k}(x^k)$ and a codifferential $\bar{w}^{m_k}$ such that $\bar{w}^{m_k}$ is the solution of subproblem 5.4.

Furthermore, either

$$\|\bar{w}^{m_k}\| \leq \delta \tag{5.10}$$

or

$$f(x^{m_k} + \lambda g^{m_k}) - f(x^{m_k}) \leq -c_1 \lambda \|\bar{w}^{m_k}\|. \tag{5.11}$$

for the search direction $g^{m_k} = -\|\bar{w}^{m_k}\|^{-1} \bar{v}^{m_k}$ holds *Step 3.* If $\|\bar{w}^{m_k}\| \leq \delta$, then **stop**. Otherwise, go to Step 4.

Step 4. Compute $x^{k+1} = x^k + \alpha_k g^{m_k}$, where α_k is defined as follows:

$$\alpha_k = \operatorname{argmax} \left\{ \alpha \geq 0 \mid \; f(x^k + \alpha g^{m_k}) - f(x^k) \leq -c_2 \alpha \|\bar{w}^{m_k}\| \right\}.$$

Set $k = k + 1$ and go to Step 2.

Theorem 5.6 *Let us assume that the function f is bounded from below*

$$f_* = \inf \{ f(x) : x \in \mathbb{R}^n \} > -\infty. \tag{5.12}$$

Then Algorithm 5.5 terminates after a finitely number $M > 0$ of iterations and generates a (λ, δ)-stationary point x^M, where

$$M \leq M_0 \equiv \left\lfloor \frac{f(x^0) - f_*}{c_2 \lambda \delta} \right\rfloor + 1.$$

Proof: Let us assume the contrary. Then, the sequence $\{x^k\}$ is infinite and points x^k are not (λ, δ)-stationary points. This means that

$$\|\bar{w}^{m_k}\| > \delta \quad (k \in \mathbb{N}).$$

Therefore, Algorithm 5.2 always finds a descent direction at each point x_k. In other words, the inequality (5.11) is satisfied. Since $c_2 \in (0, c_1]$, it follows from (5.11) that $\alpha_k \geq \lambda$. Therefore, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &< -c_2 \alpha_k \|\bar{w}^{m_k}\| \\ &\leq -c_2 \lambda \|\bar{w}^{m_k}\|. \end{aligned}$$

Since $\|\bar{w}^{m_k}\| > \delta$ for all $k \geq 0$, we get

$$f(x^{k+1}) - f(x^k) \leq -c_2\lambda\delta,$$

which implies

$$f(x^{k+1}) \leq f(x^0) - (k+1)c_2\lambda\delta$$

and, therefore, $f(x^k) \to -\infty$ $(k \to +\infty)$ which contradicts (5.12). It is obvious that the upper bound for the number M of iterations necessary to find the (λ, δ)-stationary point is M_0. △

Since $c_2 \leq c_1$, we always have $\alpha_k \geq \lambda$. Therefore $\lambda > 0$ is a lower bound for α_k, which leads to the following rule for the estimation of α_k. We define a sequence:

$$\theta_l = 2^l\lambda, \quad l = 1, 2, \ldots,$$

and α_k is the largest θ_l satisfying the inequality in Step 4 of Algorithm 5.5.

Next, we will describe the aggregated codifferential algorithm for solving Problem (5.1). Let $\{\lambda_k\}$, $\{\delta_k\}$ be sequences such that $\lambda_k \downarrow 0$, $\delta_k \downarrow 0$ as $k \to \infty$ and $\varepsilon_{opt} > 0$, $\delta_{opt} > 0$ be tolerances.

Algorithm 5.7 *A codifferential method.*

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$, and set $k = 0$.

Step 2. If $\lambda_k \leq \varepsilon_{opt}$ and $\delta_k \leq \delta_{opt}$, then stop.

Step 3. Apply Algorithm 5.5 starting from the point x^k for $\lambda = \lambda_k$ and $\delta = \delta_k$. This algorithm terminates after a finite number $M > 0$ of iterations, and as a result, it computes a (λ_k, δ_k)-stationary point x^{k+1}.

Step 4. Set $k = k + 1$ and go to Step 2.

For the point $x^0 \in \mathbb{R}^n$, we consider the set $\mathcal{L}(x^0) = \left\{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\right\}$.

Theorem 5.8 *Let us assume that f is a proper convex function and the set $\mathcal{L}(x^0)$ is bounded. Then, every accumulation point of the sequence $\{x^k\}$ generated by Algorithm 5.7 belongs to the set $X^0 = \{x \in \mathbb{R}^n \mid 0_n \in \partial f(x)\}$.*

Proof: Since the function f is proper convex and the set $\mathcal{L}(x^0)$ is bounded, $f_* > -\infty$. Therefore, conditions of Theorem 5.6 are satisfied, and Algorithm 5.5 generates a sequence of (λ_k, δ_k)-stationary points for $k \in \mathbb{N}_0$. More specifically, the point x^{k+1} is (λ_k, δ_k)-stationary, $k \in \mathbb{N}$. Then, it follows from Definition 3.4 that

$$\min\left\{\|w\| \mid w \in H(x^{k+1}, \lambda_k)\right\} \leq \delta_k. \tag{5.13}$$

It is obvious that $x^k \in \mathcal{L}(x^0)$ for $k \in \mathbb{N}_0$. The boundedness of the set $\mathcal{L}(x^0)$ implies that the sequence $\{x^k\}$ has at least one accumulation point. Let x^* be an accumulation point and $x^{k_i} \to x^*$ as $i \to +\infty$. The inequality in (5.13) implies that

$$\min\left\{\|w\| \mid w \in H(x^{k_i}, \lambda_{k_i-1})\right\} \leq \delta_{k_i-1}.$$

Then, there exists $\bar{w} \in H(x^{k_i}, \lambda_{k_i-1})$ such that $\|\bar{w}\| \leq \delta_{k_i-1}$. Considering $\bar{w} = (\bar{a}, \bar{v})$, where $\bar{v} \in \partial f(y)$ for some $y \in B_{\lambda_{k_i-1}}(x^{k_i})$, we have $\|\bar{v}\| \leq \|\bar{w}\| \leq \delta_{k_i-1}$. Therefore,

$$\min\left\{\|v\| \mid v \in \partial f(B_{\lambda_{k_i-1}}(x^{k_i}))\right\} \leq \delta_{k_i-1}.$$

Here,

$$\partial f(B_{\lambda_{k_i-1}}(x^{k_i})) = \bigcup \left\{\partial f(y) \mid \ y \in B_{\lambda_{k_i-1}}(x^{k_i})\right\}.$$

The upper semicontinuity of the subdifferential mapping $\partial f(x)$ implies that for any $\varepsilon > 0$ there exists $\eta > 0$ such that

$$\partial f(y) \subset \partial f(x^*) + B_\varepsilon(0_n) \tag{5.14}$$

for all $y \in B_\eta(x^*)$. Since $x^{k_i} \to x^*,\ \delta_{k_i}, \lambda_{k_i} \to +0$ as $i \to +\infty$, there exists $i_0 > 0$ such that $\delta_{k_i} < \varepsilon$ and

$$B_{\lambda_{k_i-1}}(x^{k_i}) \subset B_\eta(x^*)$$

for all $i \geq i_0$. Then, it follows from (5.14) that

$$\min\{\|v\| \mid \ v \in \partial f(x^*)\} \leq 2\varepsilon.$$

Since $\varepsilon > 0$ has been chosen arbitrarily, we have $0 \in \partial f(x^*)$. △

CHAPTER 6

NUMERICAL RESULTS

In this chapter, the methods mentioned in Chapters 3 - 5 ,and some other well-known methods will be compared by applying them to some academic test problems with nonsmooth objective functions. When testing new methods, the comparison is usually performed between similar kinds of methods. In other words, if a new method is a subgradient (or bundle) method, it should be compared with other subgradient (or bundle) methods. The methods developed in Chapters 3 - 5 are similar to bundle methods, but they are not exactly bundle methods since they do not store the bundle of codifferentials (or subgradient) in the memory of computer. Thus, we will use both bundle and subgradient methods in comparison. The results are analyzed using the performance profiles introduced in [29]. A short explanation for performance profile will be given in Section 6.2.

6.1 Test Problems

The well-known nonsmooth optimization academic test problems were used to test the efficiency of proposed methods by applying them to some test problems from both Chapter 2 and Chapter 3 in [61]. In numerical experiments, we do not include all test problems in [61]. The causes of excluding some problems are different. First of all, some problems, namely, CB2 and Rosen-Suzuki, are included in both Chapter 2 and 3 of [61], so in order not to repeat, we do not use them twice. The second reason is unboundedness of some problems, namely, Bard, Gamma, Colville 1 and HS78. After that, several problems, namely, PBC3, Kowalik-Osborne, EXP, PBC1, EVD61 and Filter, have more than one local solutions. After that, as the input data are not fully available for the problem TR48, we do not place the problem TR48. Lastly, the problem Transformer is not used, because of its complex coefficients. Briefly, we use 36 test problems from both Chapter 2 and 3 in [61], whereas there are 50 test problems. Although all test problems have nonsmooth objective functions, some of them are nonconvex (see Tables 6.1 - 6.2). We give brief information about the test problems in Tables 6.1 - 6.2, where the following notations are used:

- n : the number of variable of corresponding problem,
- n_A : number of functions whose maximum give objective function,
- f_{opt} : the optimal values which are reported in [61].

The problems in Chapter 2 in [61] are called *unconstrained minmax optimization problems*, whose form is as follows:

$$f(x) = \max_{1 \leq k \leq n_A} f_k(x) \quad (x \in \mathbb{R}^n). \tag{6.1}$$

Table 6.1: The brief description of unconstrained minmax problems

Problem	n	n_A	f_{opt}	Convexity
CB2	2	3	1.9522245	Convex
WF	2	3	0	Nonconvex
SPIRAL	2	3	0	Convex
EVD52	3	6	3.5997193	Convex
Rosen-Suzuki	4	4	-44	Convex
Polak6	4	4	-44	Convex
Davidon 2	4	20	115.70644	Convex
OET5	4	21	$0.26359735 \times 10^{-2}$	Convex
OET6	4	21	$0.20160753 \times 10^{-2}$	Nonconvex
Wong 1	7	5	680.63006	Convex
Wong 2	10	9	24.306209	Convex
Wong 3	20	18	93.90525	Convex
Polak 2	10	2	54.598150	Convex
Polak 3	11	10	3.70348	Convex
Watson	20	31	$0.14743027 \times 10^{-7}$	Convex
Osborne 2	11	65	$0.48027401 \times 10^{-1}$	Nonconvex

The problems in Chapter 3 in [61] are called *general unconstrained optimization problems*.

Table 6.2: The brief description of general unconstrained problems

Problem	n	f_{opt}	Convexity	Problem	n	f_{opt}	Convexity
Rosenbrock	2	0	Nonconvex	El-Attar	6	0.5598131	Nonconvex
Crescent	2	0	Nonconvex	Maxquad	10	-0.8414083	Convex
CB3	2	2	Convex	Gill	10	9.7857721	Nonconvex
DEM	2	-3	Convex	Steiner 2	12	16.703838	Nonconvex
QL	2	7.2	Convex	Maxq	20	0	Convex
LQ	2	-1.4142136	Convex	Maxl	20	0	Convex
Mifflin 1	2	-1	Convex	Goffin	50	0	Convex
Mifflin 2	2	-1	Nonconvex	MXHILB	50	0	Convex
Wolfe	2	-8	Convex	L1HILB	50	0	Convex
Shor	5	22.600162	Convex	Shell Dual	15	32.348679	Nonconvex

We test our method on aforementioned problems using 20 randomly generated starting point for each problem.

According to given tolerance $\varepsilon > 0$, if the following inequality is satisfied, it is assumed the method solves corresponding problem successfully:

$$\bar{f} - f_{opt} \leq \varepsilon(1 + |f_{opt}|),$$

where f_{opt} is the minimum value of the objective function as reported in [61] and $\bar{f}$ is the best value of the objective function found by an algorithm. In our experiments $\varepsilon = 10^{-4}$.

In our experiments, we use two subgradient methods (*SUB1* and *SUB2*), subgradient method for nonsmooth nonconvex optimization (*SUNNOPT*) and three bundle methods (*PBUN*, *PVAR* and *PNEW*). This comparison will allow us to place the proposed methods among some other nonsmooth optimization methods. Subgradient methods SUB1 and SUB2 are the subgradient method with convergent step size and the subgradient method with the constant step size, respectively, for more detail see [83]. SUNNOPT is a version of the subgradient method for general nonsmooth nonconvex optimization problems (see [44]). Subroutine PBUN is based on the proximal bundle method [55, 64, 65, 62]. PVAR is the variable metric method in [59, 57, 85]. The last bundle method, PNEW, is the bundle-Newton method in [58].

In the implementation, we needed to set parameters in Algorithms 3.8, 4.7 and 5.7. They were chosen as follows: $c_1 = 0.2$, $c_2 = 0.05$, $\delta_k \equiv \delta_{opt} = 10^{-7}$, $\lambda_1 = 1$ and $\varepsilon_{opt} = 10^{-10}$ in all algorithms. In Algorithms 3.8 and 4.7, we used $\lambda_{k+1} = 0.1\lambda_k$ $(k \geq 1)$. In Algorithm 5.7, we referred to $\lambda_{k+1} = 0.5\lambda_k$ $(k \geq 1)$. The parameters for the methods SUB1 and SUB2 were chosen as the following rule:

$$x_{k+1} = x_k - t_k \xi_k \quad (k \in \mathbb{N}_0$$

SUB1: We used the step-length $t_k = 1/k$ for first 25000 iteration, and after that we update it as the following rule in order to improve the convergence of the algorithm. Let p_k be the largest integer such that $p_k \leq \dfrac{k}{25000}$. We put

$$t_k = \frac{1}{k - p_k}.$$

SUB2: We used $t_k = 0.0005$ for the first 10000 iterations, and $t_k = 0.0001$ for all other iterations.

The parameters for the SUNNOPT were $c_1 = 0.2$, $c_2 = 0.05$, $h_{j+1} = 0.8h_j (j \in \mathbb{N})$, $h_1 = 1$, $\delta_j \equiv 10^{-7} (j \in \mathbb{N}_0)$. Although convergence of the subgradient methods was proved only for convex functions in [83], we applied them also to nonconvex problems in this thesis. The subgradient methods do not have any stopping criterion. Therefore, in our experiments, the algorithm terminates when the number of function evaluations reaches 10^6 or if it cannot decrease the value of the objective function with respect to tolerance 10^{-4} in 5000 successive iterations.

6.2 Performance Profiles

In this section, the idea behind performance profiles will be given briefly (for more detail, see [29]). n_p denote the number of problems. As a performance measure, computing time, the number of function evaluations (or subgradient evaluations) and the number of iterations, etc., can be used. Since the computational times of methods are very small, we will be interested in the number of function evaluations and subgradient evaluations. The following idea can be applied to other measures. For each problem p and solver s, we need the data $n_{p,s}$ which is the number of function (or subgradient) evaluations required to solve problem p by solver s.

On fixed problem p, we aim to compare the performance of solver s with the best performance of any solver. In other words, we use the following performance ratio in comparison

$$r_{p,s} = \ln \frac{n_{p,s}}{\min\{n_{p,s} : s \in S\}},$$

where S is the set of all solvers. Here, the performance ratio $r_{p,s}$ is in logarithmic (or ln) scale. We define a parameter $r_M \geq r_{p,s}$ for all problems p and solvers s. The equality holds if and only if solver s

does not solve problem p. “The choice of r_M does not affect the performance evaluation.” was shown in [29]. We interest in the performance of solver s on all problems, so the probability $\rho_s(\tau)$ is defined as follows:

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

where “size= $\{p \in P : r_{p,s} \leq \tau\}$” denotes the number of problems such that $r_{p,s} \leq \tau$. Obviously, ρ_s is the cumulative distribution function for the performance ratio.

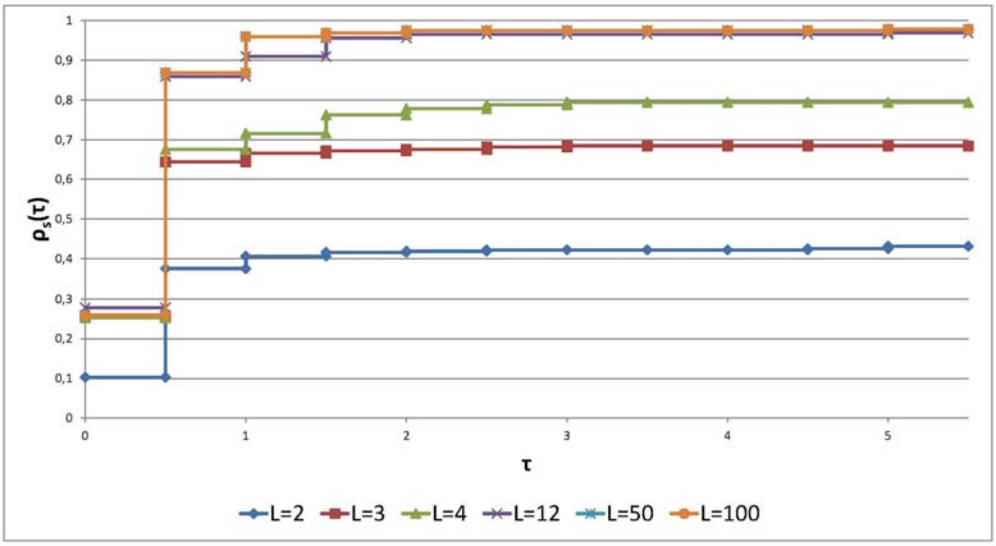
The performance profile $\rho_s : \mathbb{R} \to [0, 1]$ for a solver is a nondecreasing, piecewise constant function and continuous from the right. The initial value of $\rho_s(\tau)$ (i.e., $\rho_s(0)$) shows the percentage of test problems for which the corresponding method uses least evaluations. The value of $\rho_s(\tau)$ at the end of abscissa gives the percentage of the test problems solved successfully by the corresponding method, i.e., the reliability of the solver (this does not depend on the measured performance). In addition, the relative efficiency of each solver can be directly observed from the performance profiles by distance in hight term between curves. This measure shows us how much better the corresponding solver.

6.3 Results

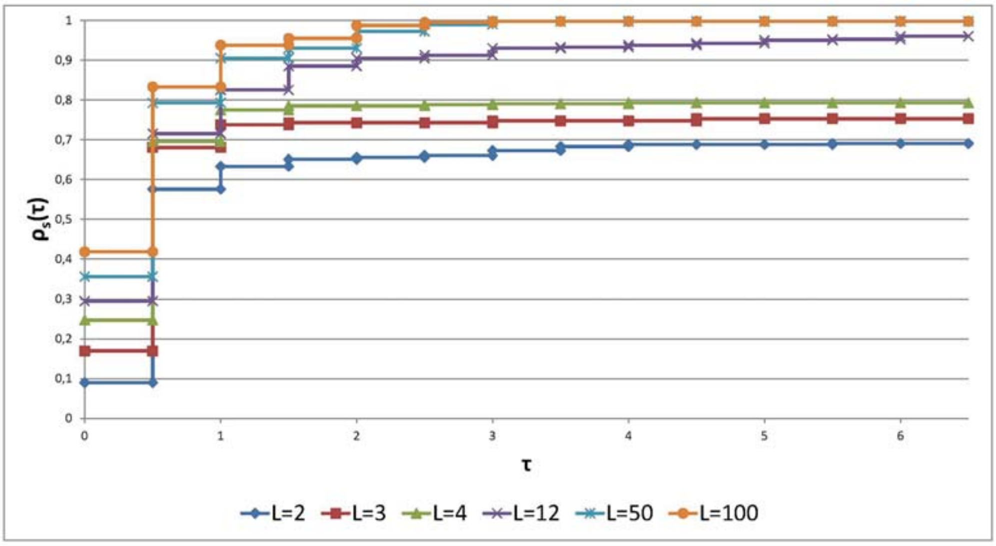
In this section, first of all, we give some numerical results for *Aggregate Codifferential Method (ACM)* with the number $l = 2, 3, 4, 12, 50, 100$. These results are analyzed using the performance profile idea. After that, numerical results for our other two methods and ACM with $l = 12$ are compared with some subgradient and bundle methods. The reasons for why subgradient and bundle methods are chosen for comparisons are different. First, the reason of choosing subgradient methods is the similarity between the proposed methods and subgradient methods with respect to construction of methods. Secondly, using bundle methods for comparisons is based on completely different reason. Bundle methods are the best methods in literature and according to specialists, they are assumed to be more complex and advanced than subgradient methods, so they are chosen for comparison in order to replace proposed methods in literature.

6.3.1 Results of ACM With the Number $l = 2, 3, 4, 12, 50$ and 100

The following performance profiles in Figures 6.1 - 6.2 demonstrate the differences among ACM with $l = 2, 3, 4, 12, 50$ and 100. We can observe that our proposed method ACM solves problems more successfully when the number l became bigger. In Figure 6.1(b), it can be understood that the number of function evaluation decreases while the number l is increasing; however, in Figure 6.1(a), we can not observe the same situation. This means the relation between the number l and the number of function evaluations depends on the types of problems.

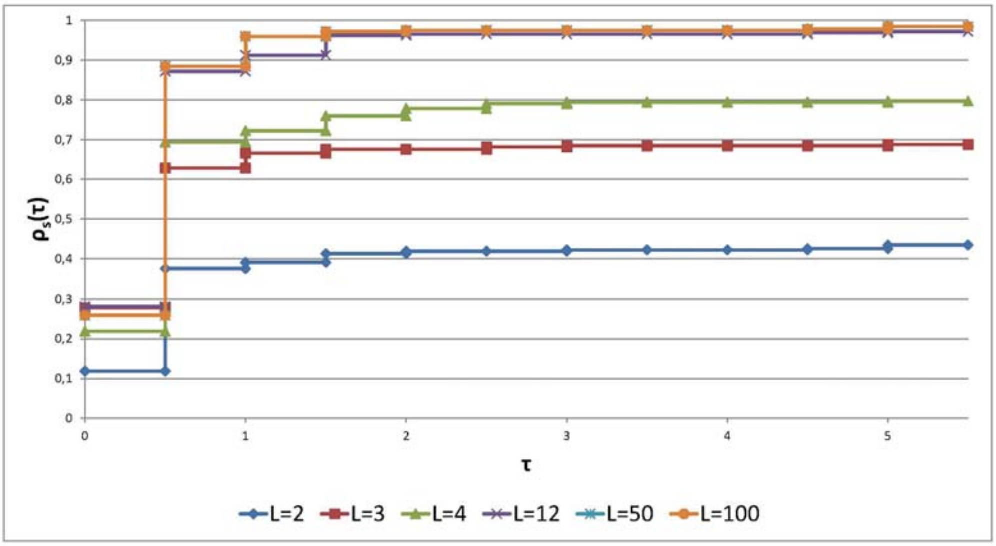


(a) For Unconstrained Minmax Optimization Problems

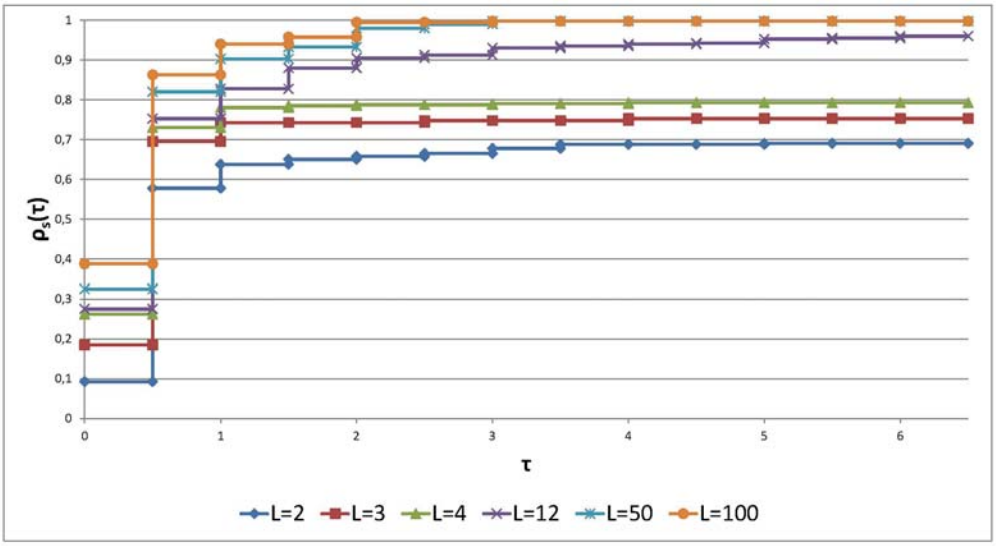


(b) For General Unconstrained Optimization Problems

Figure 6.1: Performance Profile Graphs with respect to the Number of Function Evaluation, Given the Tolerance $\varepsilon = 10^{-4}$



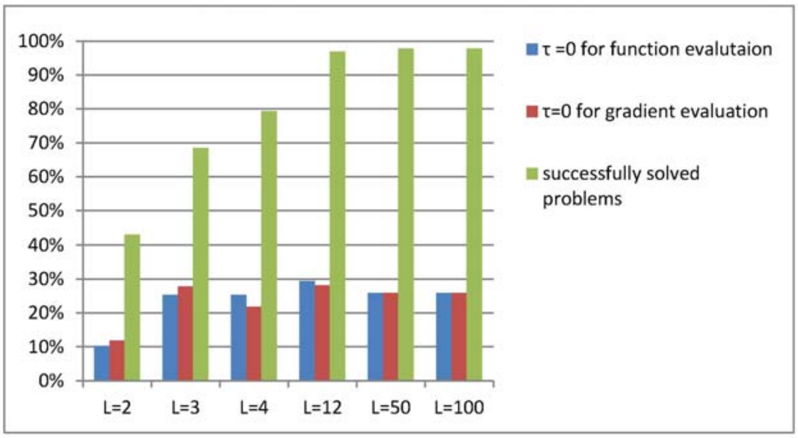
(a) For Unconstrained Minmax Optimization Problems



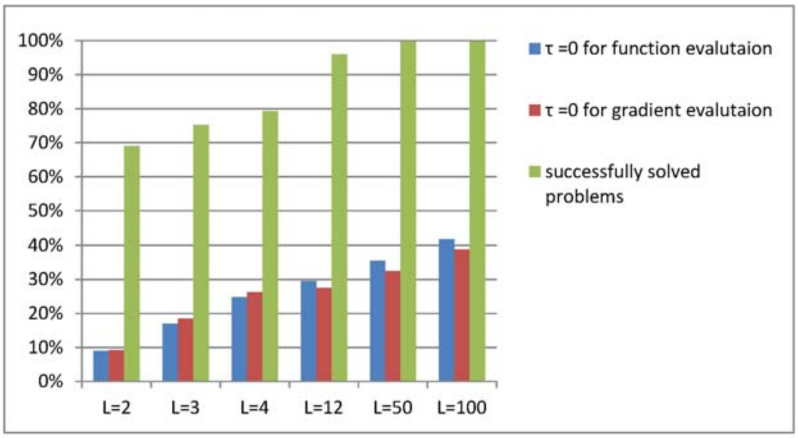
(b) For General Unconstrained Optimization Problems

Figure 6.2: Performance Profile Graphs with respect to the Number of Gradient Evaluation, Given the Tolerance $\varepsilon = 10^{-4}$

In order to observe better, Figure 6.3 is put. In Figure 6.3, the first two columns shows the percentage of test problems for which the corresponding method uses the smallest function and gradient evaluations, respectively. The last columns show the percentage of the test problems solved successfully by the corresponding method.



(a) Unconstrained Minmax Optimization Problems



(b) General Unconstrained Optimization Problems

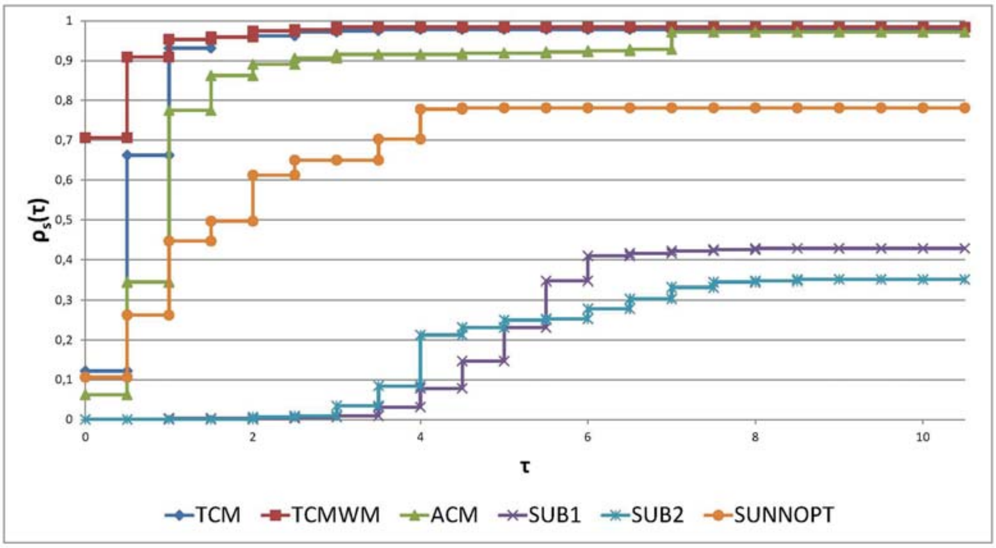
Figure 6.3: Column Charts for ACM with $l = 2, 3, 4, 12, 50$ and 100

In Figure 6.3(a), we can easily observe that the proposed methods have a similar accuracy when $l \geq$

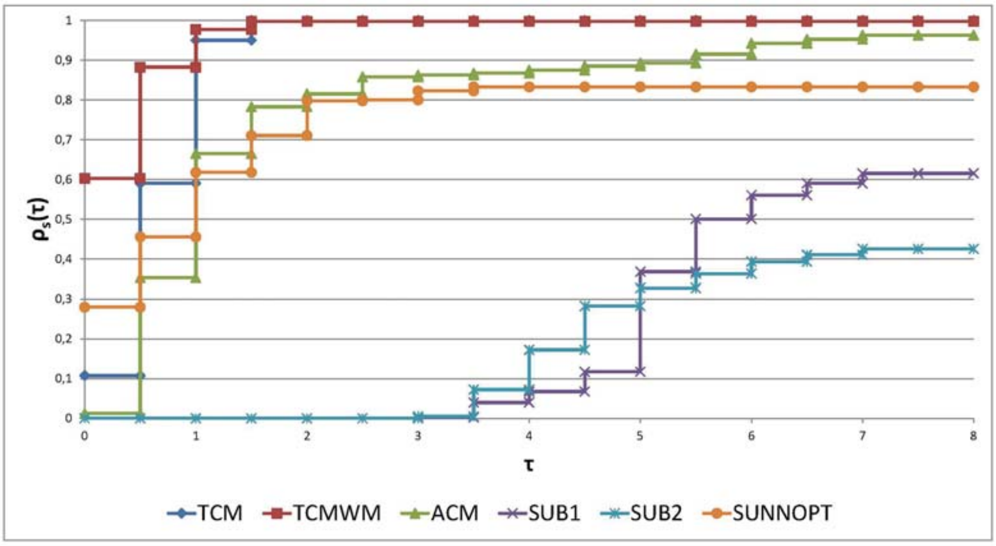
12, and the percentages of successfully solved problems are bigger than 95%. On the other hand, the numbers of function and gradient evaluation are directly proportional with the number l when $l \geq 12$. In Figure 6.3(b), the percentages of the proposed methods with $l = 50$ and $l = 100$ are 100%. The numbers of function and gradient evaluation are in an inverse proportion for the general unconstrained nonsmooth problems; this situation can be also observed from Figures 6.1(b) and 6.2(b). As a conclusion, we can say that the proposed methods became more efficient when the number l increases for all type problems, and the number of function and gradient evaluations change by a different variety, in other words, they do not directly depend on the number l.

6.3.2 Comparisons for Proposed Methods

In this section, it can be found eight different performance profile graphs and four column charts. The first four performance profile graphs are to compare our proposed methods and subgradient methods with respect to function and subgradient evaluation for unconstrained minmax and general nonsmooth problems. After this comparison with subgradient methods, the same kind of comparison is done with bundle methods in the next four performance profile graphs. Former comparisons show us the differences between our proposed methods and other subgradient methods with respect to the number of function and subgradient evaluations. One can observe that our proposed methods use less function and subgradient evaluations than subgradient methods. About the reliability of our proposed methods, there is a big difference between the proposed methods and subgradient methods, so it is needed to compare them with bundle methods, which are known as the best methods in literature. Thus, the latter comparison has been done to determine accuracy of the proposed methods. Column charts, which are provided for both comparisons, can be considered as a brief explanations, together with statistic values.

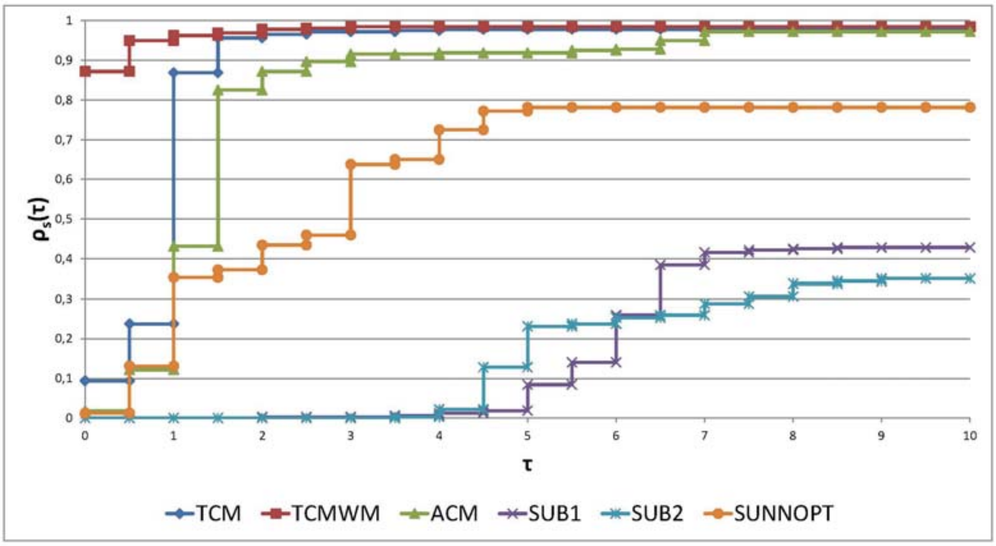


(a) For Unconstrained Minmax Optimization Problems

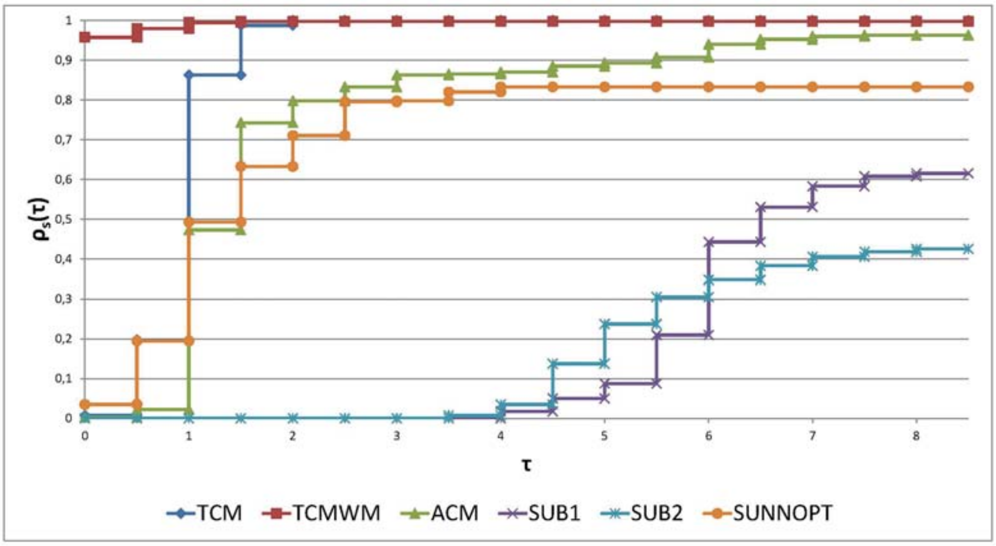


(b) For General Unconstrained Nonsmooth Optimization Problems

Figure 6.4: Comparison of the Proposed Methods with Subgradient Methods with respect to the Number of Function Evaluation, Given the Tolerance $\varepsilon = 10^{-4}$



(a) For Unconstrained Minmax Optimization Problems



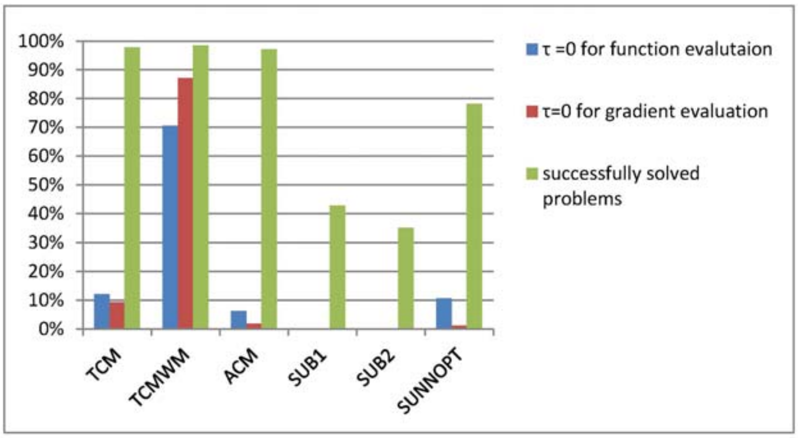
(b) For General Unconstrained Nonsmooth Optimization Problems

Figure 6.5: Comparison of the Proposed Methods with Subgradient Methods with respect to the Number of Subgradient Evaluation, Given the Tolerance $\varepsilon = 10^{-4}$

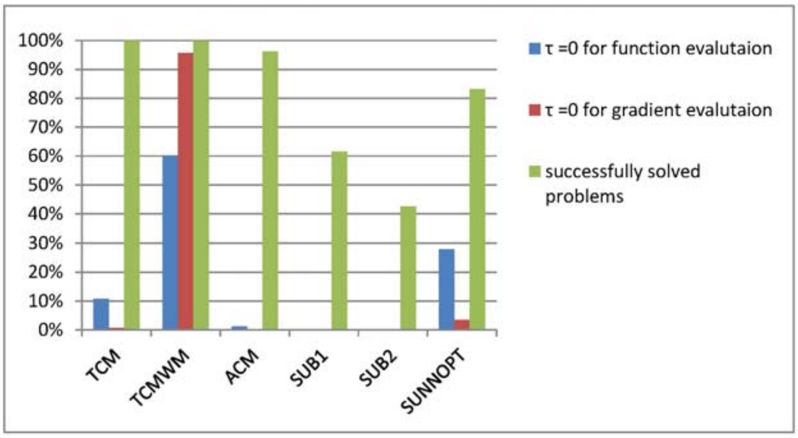
From Figures 6.4 - 6.5, it can be observed that the proposed methods have a higher accuracy than subgradient methods, for both unconstrained minmax and general nonsmooth problems. Actually, TCM and TCMWM solve all general nonsmooth unconstrained problems without depending on starting points, in other words, they solved 100% of that problems for all the given starting points. Here, it should also be highlighted that the percentage of successfully solved problems of ACM for general nonsmooth problems is very close to those of TCM and TCMWM, respectively. For unconstrained minmax problems the results of TCM, TCMWM and ACM demonstrate that the successes of these methods are almost equivalent. This situation can be observed from Figures 6.4(b) and 6.5(b) where the percentage of successfully solved problems is almost a hundred.

From Figure 6.5, we can easily learn that TCMWM uses the least number of gradient evaluation when comparing not only with subgradient methods but also with our other methods. Thus, we can say that our aim to improve from TCM to TCMWM is achieved.

The following column charts (Figure 6.6) are just put to observe easily differences among the percentages of the least function and gradient evaluations (blue and red bars respectively), and the percentage of successfully solved problems (green bar). From Figure 6.6, it is obvious that the proposed methods reach accurately optimum values by using less function and gradient evaluations.



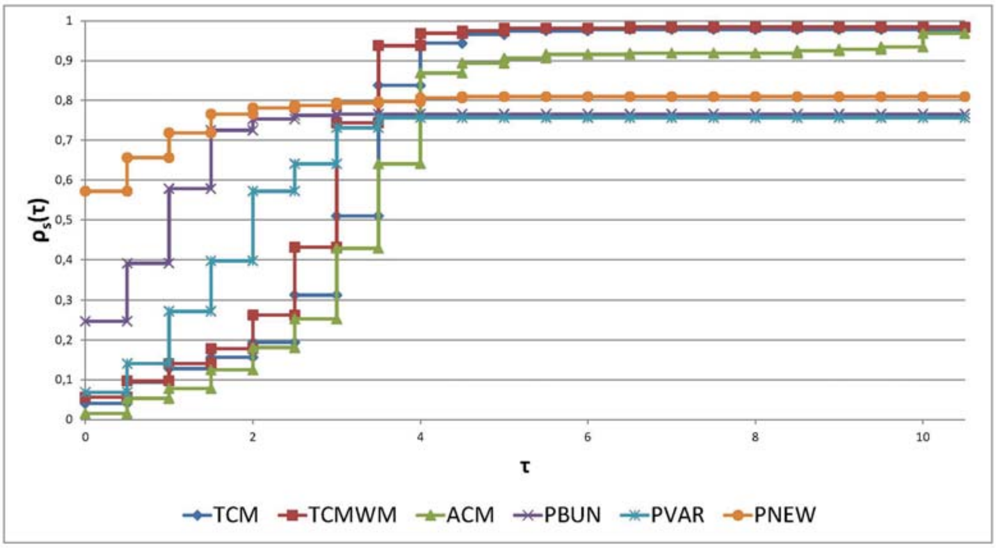
(a) Unconstrained Minmax Optimization Problems



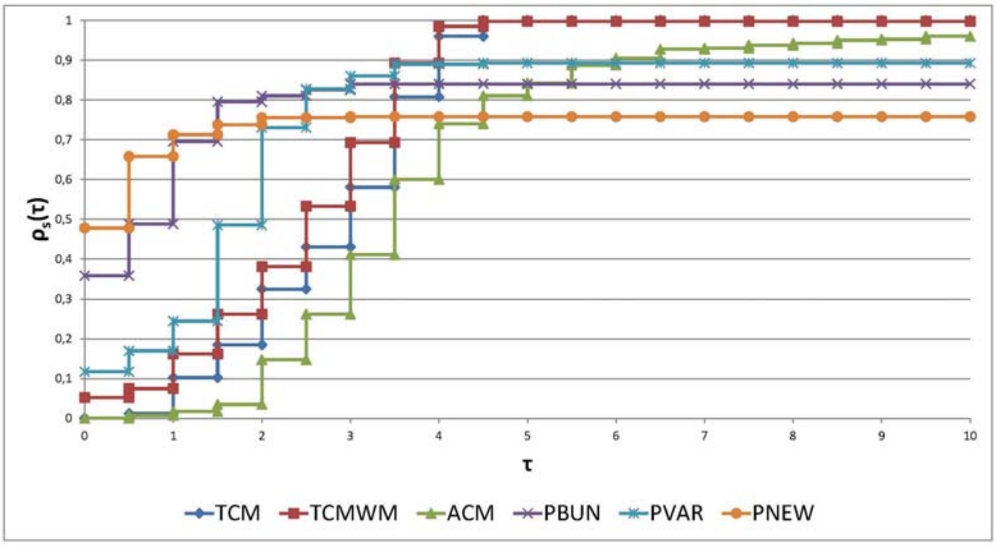
(b) General Unconstrained Nonsmooth Optimization Problems

Figure 6.6: Column Charts for the Proposed Methods in order to Compare with Subgradient Methods

The following four performance profile graphs are prepared for a comparison between the proposed methods and bundle methods. As known, bundle methods use very few function and subgradient evaluations, so the gap between the proposed methods and the bundle methods in Figures 6.7 - 6.8 at the beginning of that graphs is acceptable. Towards the end of the graphs, the proposed methods catch up bundle methods, it means after some value of the ratio of function (or subgradient) evaluations, proposed methods solve more problems. Consequently, it is important that the proposed methods have became better than the other methods towards the end of the graphs in Figures 6.7 - 6.8, since bundle methods have the highest accuracy in the literature. In other words, our proposed methods are more accurate than bundle methods.

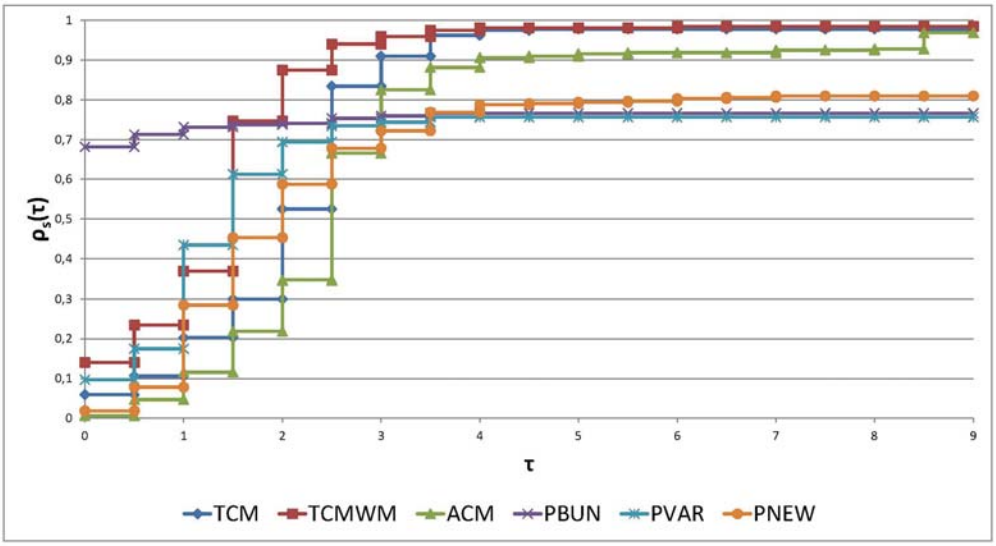


(a) For Unconstrained Minmax Optimization Problems

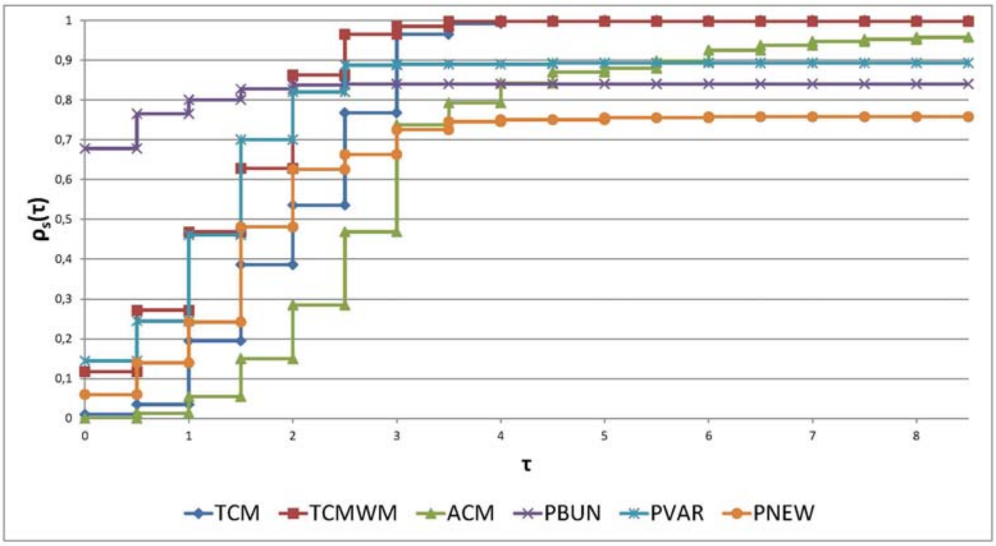


(b) For General Unconstrained Nonsmooth Optimization Problems

Figure 6.7: Comparison of the Proposed Methods with Bundle Methods with respect to the Number of Function Evaluation,Given the Tolerance $\varepsilon = 10^{-4}$

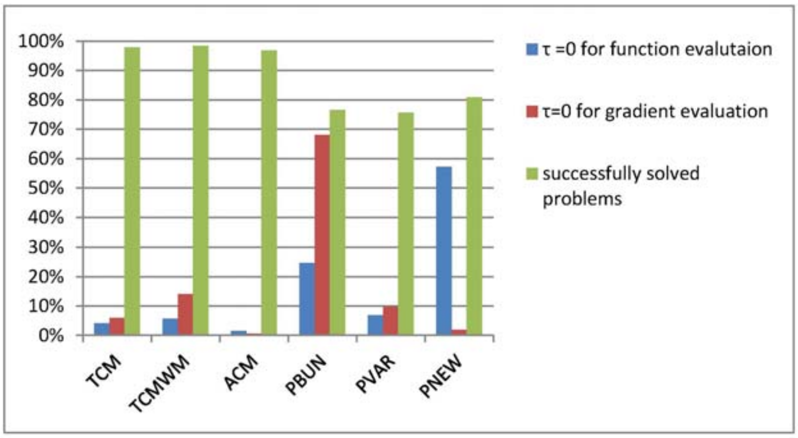


(a) For Unconstrained Minmax Optimization Problems

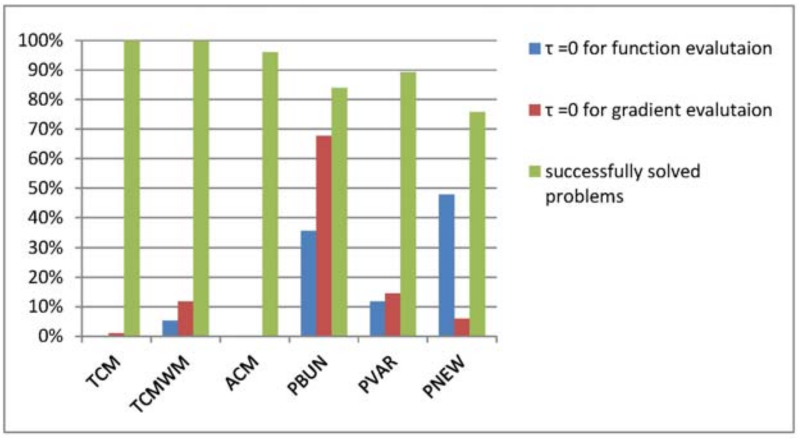


(b) For General Unconstrained Nonsmooth Optimization Problems

Figure 6.8: Comparison of the Proposed Methods with Bundle Methods with respect to the Number of Subgradient Evaluation, Given the Tolerance $\varepsilon = 10^{-4}$



(a) Unconstrained Minmax Optimization Problems



(b) General Unconstrained Nonsmooth Optimization Problems

Figure 6.9: Column Chart for the Proposed Methods in order to Compare with Bundle Methods

Consequently, all performance profile graphs and column charts in this chapter display that our proposed methods are more robust than both subgradient methods and bundle methods, so they can be considered as an alternative methods for nonsmooth convex optimization problems. In this chapter, we also test our methods for nonsmooth nonconvex problems, although they were not developed for the nonconvex case. The results for nonconvex case show us that these methods can be used for them. Maybe, the generalization of these methods can be considered as an open problem.

CHAPTER 7

TRUNCATED CODIFFERENTIAL METHOD FOR LINEARLY CONSTRAINED NONSMOOTH OPTIMIZATION PROBLEMS

In this chapter, a new algorithm is developed to minimize linearly constrained non-smooth optimization problem for convex objective functions. The algorithm is based on the concept of the codifferential. The convergence of the proposed minimization algorithm is proved and results of numerical experiments are reported using a set of test problems with nonsmooth convex objective functions.

7.1 Introduction

In this chapter, We focus on the solution of the following linearly constrained nonsmooth optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array} \tag{7.1}$$

where $X = \left\{x \in \mathbb{R}^n \,\middle|\, A_1 x = b^1, A_2 x \leq b^2\right\}$ such that A_1 is an $p_1 \times n$, A_2 is a $p_2 \times n$ matrix, $b^1 \in \mathbb{R}^{p_1}$, $b^2 \in \mathbb{R}^{p_2}$ and it is assumed that the objective function f is convex. There are many methods to solve the unconstrained minimization problem ($p_1 = 0$ and $p_2 = 0$), namely, subgradient methods [83] and different versions of the bundle methods [33, 35, 36, 43, 47, 49, 55]. By using the concept of the codifferential, the truncated codifferential method [12] has been developed for unconstrained problem. In order to solve the problem (7.1), some methods have also been developed [13, 56].

7.2 Linearly Constrained Nonsmooth Optimization Problems

The problem (7.1) can be reduced to a linearly equality constrained optimization problem by introducing a slack variable:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array} \tag{7.2}$$

where $X = \{x \in \mathbb{R}^n \,|Ax = b\}$ such that A is an $m \times n$ matrix, $b \in \mathbb{R}^m$ and it is assumed that the objective function is convex. Without loss of generality we assume that the rank of matrix A is equal to $m < n$. We can split variables $x_1, ..., x_n$ into two parts: $x^T = (x_B^T, x_N^T)$ where $x_B \in \mathbb{R}^{n-m}$ is called a vector of *basic (or independent)* variables and $x_N \in \mathbb{R}^m$ is a vector of *non-basic (or dependent)* variables. Then the matrix A can be partitioned as follows:

$$A = (A_B, A_N),$$

where A_B is an $m \times (n-m)$ matrix consisting of columns related with x_B of the matrix A and A_N is an $m \times m$ matrix consisting of columns related with x_N of the matrix A. Here, A_N is an invertible matrix. The equality constraint system can be rewritten as follows:

$$A_B x_B + A_N x_N = b.$$

This system of linear equations can be solved with respect to non-basic variable x_N:

$$x_N = A_N^{-1}\left(b - A_B x_B\right).$$

Thus, we can represent the non-basic variables x_N as follows:

$$x_N = Bx_B + c \text{ where } B = -A_N^{-1}A_B \text{ and } c = A_N^{-1}b.$$

Then, the objective function f can be rearranged with respect to basic variables x_B and non-basic variables x_N as

$$f(x) = f((x_B^T, x_N^T)^T) = f((x_B^T, (Bx_B + c)^T)^T).$$

According to the above rearrangement, we can define the following function

$$h(y) = f((y^T, (By + c)^T)^T) \quad (y \in \mathbb{R}^{n-m}).$$

Proposition 7.1 *Assume that the function f is a convex function on $\mathbb{R}^n$. Then the function $h(y) = f\left((y^T, (By + c)^T)^T\right)$ is convex, where $B = -A_N^{-1}A_B$ and $c = A_N^{-1}b$.*

Proof: One can evaluate as follows:

$$\begin{aligned}
h(\lambda y_1 + (1-\lambda)y_2) &= f\left(\left(\lambda y_1 + (1-\lambda)y_2\right)^T, \left(B\left[\lambda y_1 + (1-\lambda)y_2\right] + c\right)^T\right)^T\Big) \\
&= f\left(\left(\lambda y_1^T + (1-\lambda)y_2^T, \lambda(By_1 + c)^T + (1-\lambda)(By_2 + c)^T\right)^T\right) \\
&= f\left(\left(\lambda y_1^T, \lambda(By_1 + c)^T\right)^T + \left((1-\lambda)y_2^T, (1-\lambda)(By_2 + c)^T\right)^T\right) \\
&= f\left(\lambda\left(y_1^T, (By_1 + c)^T\right)^T + (1-\lambda)\left(y_2^T, (By_2 + c)^T\right)^T\right) \\
&= f\left(\lambda x_1 + (1-\lambda)x_2\right), \text{ where } x_1 = \left(y_1^T, (By_1 + c)^T\right)^T \text{ and } x_2 = \left(y_2^T, (By_2 + c)^T\right)^T \\
&\leq \lambda f(x_1) + (1-\lambda)f(x_2), \text{ since } f \text{ is convex} \\
&= \lambda f\left(\left(y_1^T, (By_1 + c)^T\right)^T\right) + (1-\lambda)f\left(\left(y_2^T, (By_2 + c)^T\right)^T\right) \\
&= \lambda h(y_1) + (1-\lambda)h(y_2)
\end{aligned}$$

$\Rightarrow$ h is convex.

$\triangle$

Now, we can consider the following unconstrained optimization problem,

$$\begin{array}{ll} \text{minimize} & h(y) \\ \text{subject to} & y \in \mathbb{R}^{n-m} \end{array} \tag{7.3}$$

Proposition 7.2 *Let $y^* \in \mathbb{R}^{n-m}$ be a solution of the problem (7.3). Then, $x^* = \left((y^*)^T, (By^* + c)^T\right)^T$ is a solution of the problem (7.2), where $B = -A_N^{-1}A_B$ and $c = A_N^{-1}b$.*

Proof: Let $y^* \in \mathbb{R}^{n-m}$ be a stationary point of the problem (7.3). First, let us show that $x^* = \left((y^*)^T, (By^*+c)^T\right)^T \in X = \{x \in \mathbb{R} \,| Ax = b\}$. Using partitioned matrix and vector, we get the follows:

$$\begin{aligned} Ax^* &= (A_B, A_N)\left((y^*)^T, (By^*+c)^T\right)^T = A_B y^* + A_N(By^*+c) = A_B y^* + A_N B y^* + A_N c) \\ &= A_B y^* + A_N(-(A_N)^{-1}A_B)y^* + A_N((A_N)^{-1}b) \text{ ,since } B = -A_N^{-1}A_B \text{ and } c = A_N^{-1}b \\ &= A_B y^* - A_B y^* + b = b \\ \Rightarrow \quad & x^* \in X. \end{aligned}$$

Now we prove that x^* is a solution of the problem (7.2),

$$\begin{aligned} & \min_{y\in\mathbb{R}^{n-m}} \{h(y)\} = h(y^*) \\ \Rightarrow \quad & \min_{y\in\mathbb{R}^{n-m}} \left\{ f\left(\left(y^T, (By+c)^T\right)^T\right)\right\} = f\left(\left((y^*)^T, (By^*+c)\right)^T\right) \\ \Rightarrow \quad & \min_{x\in X}\{f(x)\} = f\left(\left((y^*)^T, (By^*+c)^T\right)^T\right), \text{ since } X = \left\{x \in \mathbb{R}^n \,\middle|\, x^T = \left(y^T, (By+c)^T\right) \ \forall y \in \mathbb{R}^{n-m}\right\}. \end{aligned}$$

△

Proposition 7.3 *Let $x^* \in X$ be a solution of the problem (7.2). Then, there exists $y^* \in \mathbb{R}^{n-m}$ such that $x^* = \left((y^*)^T, (By^*+c)^T\right)^T$ and y^* is a solution of problem (7.3), where $B = -A_N^{-1}A_B$ and $c = A_N^{-1}b$.*

Proof: The existence of the y^*: Consider $x^* \in X$, as mentioned before we can divide variable $x_1^*, ...x_n^*$ into two parts: $(x^*)^T = ((x_B^*)^T, (x_N^*)^T)$ as a vector of basic variables and nonbasic variables. Clearly $x^* = \left((x_B^*)^T, (x_N^*)^T\right)^T = \left((y^*)^T, (By^*+c)^T\right)^T$ so $y^* = x_B^*$. Now, we prove that y^* is a solution of Problem (7.3):

$$\begin{aligned} & \min_{x\in X}\{f(x)\} = f(x^*) \\ \Rightarrow \quad & \min_{x_B\in\mathbb{R}^{n-m}} \left\{ f\left(\left(x_B^T, x_N^T\right)^T\right)\right\} = f\left(\left((x_B^*)^T, (x_N^*)^T\right)^T\right), \text{ where } x_N = Bx_B + c \\ \Rightarrow \quad & \min_{x_B\in\mathbb{R}^{n-m}} \{h(x_B)\} = h(x_B^*) \\ \Rightarrow \quad & \min_{y_B\in\mathbb{R}^{n-m}} \{h(y_B)\} = h(y^*), \text{ since } y^* = x_B^* \end{aligned}$$

△

As the result of Proposition (7.2) and (7.3), Problem (7.2) can be reduced to the unconstrained minimization Problem (7.3).

7.3 Computation of a Descent Direction

In this section, our aim is to compute descent direction of Problem (7.2). To reach this aim, first, we compute a descent direction of Problem (7.3). The following subset of the hypodifferential is sufficient to find such directions. For any given $\lambda \in (0,1)$, let

$$H(y,\lambda) := cl\ co \left\{ w = (a,v) \in \mathbb{R}\times\mathbb{R}^{n-m} \,\middle|\, \begin{array}{l} \exists\ \bar{y} \in B_\lambda(y), \\ v \in \partial h(\bar{y}), \\ a = h(\bar{y}) - h(y) - \langle v, \bar{y} - y\rangle \end{array} \right\}. \tag{7.4}$$

It is clear that $a \le 0$ for all $w = (a,v) \in H(y,\lambda)$. Since $a = 0$ at the point y,

$$\max_{w=(a,v)\in H(y,\lambda)} a = 0. \tag{7.5}$$

If $\bar{B}_\lambda(x) \subset \text{int}U$ for all $\lambda \in (0,1)$ where $U \subset \mathbb{R}^n$ is a closed convex set, then from the definition of both the hypodifferential and the set $H(x,\lambda)$, the following inclusion holds:

$$H(y,\lambda) \subset \underline{d}h(y) \quad \forall\, \lambda \in (0,1).$$

The sets $H(y,\lambda)$ is called *truncated codifferentials* of the function h at the point y.

Proposition 7.4 *Let us assume that* $0_{n-m+1} \notin H(y,\lambda)$ *for a given* $\lambda \in (0,1)$ *and*

$$\|w^0\| = \min\{\|w\| \mid \ w \in H(y,\lambda)\} > 0.$$

$w^0 = (a_0, v^0)$. *Then,* $v^0 \neq 0_n$ *and*

$$h(y + \lambda g^0) - h(y) \leq -\lambda\|v^0\|, \tag{7.6}$$

where $g^0 = -\|w^0\|^{-1}v^0$.

The proof of Proposition 7.4 can be done as the proof of Proposition 3.1. Proposition 7.4 implies that the set $H(y,\lambda)$ can be used to find descent directions of the the function h. Because of the difficulty of computation $H(y,\lambda)$, the following algorithm is developed to find descent direction by using a few elements of $H(y,\lambda)$ as Algorithm 7.5.

Algorithm 7.5 Computation of descent directions.

Step 0. Let the numbers $\lambda \in (0,1)$, $c \in (0,1)$ and a sufficiently small number $\delta > 0$ be given.

Step 1. Chose any $g^1 \in S_1$, and compute $v^1 \in \partial h(y + \lambda g^1)$ and $a_1 = h(y+\lambda g^1) - h(y) - \lambda\langle v^1, g^1\rangle$. Set $\bar{H}_1(y) = \{w^1 = (a_1, v^1)\}$ and $k = 1$.

Step 2. Compute the $\bar{w}^k = (\bar{a}_k, \bar{v}^k) \in \mathbb{R} \times \mathbb{R}^{n-m}$ solving the quadratic subproblem:

$$\min \quad \|w\|^2 \quad \text{such that} \quad w \in \bar{H}_k(y). \tag{7.7}$$

Step 3. If

$$\|\bar{w}^k\| \leq \delta, \tag{7.8}$$

then **stop**. Otherwise, compute $\bar{g}^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ and go to Step 4.

Step 4. If

$$h(y + \lambda\bar{g}^k) - h(y) \leq -c\lambda\|\bar{w}^k\|, \tag{7.9}$$

then **stop**. Otherwise, set $g^{k+1} = \bar{g}^k$ and go to Step 5.

Step 5. Compute $v^{k+1} \in \partial h(y + \lambda g^{k+1})$ and $a_{k+1} = h(y + \lambda g^{k+1}) - h(y) - \lambda\langle v^{k+1}, g^{k+1}\rangle$. Construct the set $\bar{H}_{k+1}(y) = \text{co}\,\{\bar{H}_k(y) \bigcup \{w^{k+1} = (a_{k+1}, v^{k+1})\}\}$, set $k \leftarrow k+1$ and go to Step 2.

Stated briefly, the above mentioned algorithm works as follows; in Step 1 a direction g^1 is selected and the element of the truncated codifferential in this direction is computed. In Step 2, the smallest length of the truncated codifferential from the convex hull is found. This is a quadratic problem and to solve it, there are several methods [32, 87]. In Step 3, we check whether this smallest length is less than give tolerance $\delta > 0$, or not. If it holds, we reach the approximate stationery point. Otherwise, we compute an other search direction. In Step 4, if the currently computed search direction satisfies (7.9), then the algorithm stops. In Step 5, a new element of the truncated codifferential in the direction g^{k+1} is computed. This algorithm terminates after finitely many steps by Proposition 3.3, because h is a proper convex function.

7.4 A Truncated Codifferential Method

In this section, we describe the truncated codifferential method for solving problem (7.3). We are able to compute the solution of problem (7.2) according to Proposition 7.2.

Definition 7.6 *A point $y \in \mathbb{R}^{n-m}$ is called a (λ, δ)-stationary point of the function h if*

$$\min_{w \in H(y,\lambda)} \|w\| \leq \delta.$$

The (λ, δ)-stationary point of the function h will be computed by Algorithm 7.7.

Algorithm 7.7 *The truncated codifferential method for finding (λ, δ)-stationary points.*

Step 0. Let $\lambda \in (0, 1),\ \delta > 0,\ c_1 \in (0, 1), c_2 \in (0, c_1]$ be given numbers.

Step 1. Start with any point $y^0 \in \mathbb{R}^{n-m}$ and set $k = 0$.

Let $\lambda \in (0, 1),\ \delta > 0,\ c_1 \in (0, 1)$ and $c_2 \in (0, c_1]$ be given numbers.

Step 2. Apply Algorithm 7.5 setting $y = y^k$. This algorithm terminates after finite number of iterations. Thus, we have the set $\bar{H}_m(y^k)$ and an element $\bar{w}^k$ such that

$$\|\bar{w}^k\|^2 = \min\left\{\|w\|^2 \mid\ w \in \bar{H}_m(y^k)\right\}.$$

Furthermore, either

$$\|\bar{w}^k\| \leq \delta \tag{7.10}$$

or

$$h(y^k + \lambda g^k) - h(y^k) \leq -c_1\lambda\|\bar{w}^k\|. \tag{7.11}$$

for the search direction $g^k = -\|\bar{w}^k\|^{-1}\bar{v}^k$ holds.

Step 3. If $\|\bar{w}^k\| \leq \delta$, then **stop**. Otherwise, go to Step 4.

Step 4. Compute $y^{k+1} = y^k + \alpha_k g^k$, where α_k is defined as follows:

$$\alpha_k = \text{argmax}\ \left\{\alpha \geq 0 \mid h(y^k + \alpha g^k) - h(y^k) \leq -c_2\alpha\|\bar{w}^k\|\right\}.$$

Set $k \leftarrow k + 1$ and go to Step 2.

Without loss of generality, we can assume the function h is bounded from below, so that Algorithm 7.7 terminates after finitely many step because of Theorem 3.6. We will describe Algorithm 7.8 in order to solve Problem (7.3).

Algorithm 7.8 *The truncated codifferential method.*

Step 0. Let λ_k, δ_k be sequance such that $\lambda_k \to 0$ and $\delta_k \to 0$ as $k \to 0$.

Step 1. Choose any starting point $y^0 \in \mathbb{R}^{n-m}$, and set $k = 0$.

Step 2. If $0_{n-m} \in \partial h(x^k)$, then stop.

Step 3. Apply Algorithm 7.7 starting from the point y^k for $\lambda = \lambda_k$ and $\delta = \delta_k$. This algorithm terminates after a finite number of iterations $M > 0$, and as a result, it computes a (λ_k, δ_k)-stationary point y^{k+1}.

Step 4. Set $k \leftarrow k + 1$ and go to Step 2.

Every accumulation point of the sequence y_k generated by Algorithm 7.8 is the solution of Problem (7.3) according to Theorem 3.9, because the function h is convex. Let y^* be a solution of Problem (7.3) which is obtained from Algorithm 7.8. According to Proposition 7.2, the solution of problem (7.2) x^* is as follows:

$$(x^*)^T = ((y^*)^T, (By^* + c)^T), \text{ where } B = -A_N^{-1}A_B \text{ and } c = A_N^{-1}b.$$

7.5 Examples

In order to check the efficiency of the method, we use three test problems. In all examples, each slack variable is assigned as a basic variable.

Example 7.9 *(Problem 1 in [13])*

$$\begin{array}{ll} \textit{minimize} & f(x) \\ \textit{subject to} & 2x_1 - x_2 + x_3 - x_4 = 1 \end{array} \tag{7.12}$$

where $f(x) = |x_1-1|+100|x_2-|x_1||+90|x_4-|x_3||+|x_3-1|+10.1(|x_2-1|+|x_4-1|)+4.95(|x_2+x_4-2|-|x_2-x_4|)$.

We divide the variables x_1, x_2, x_3 and x_4 into two parts as $x^T = ((x_B)^T, x_N) = (x_1, x_2, x_3, x_4)$. The relationship between basic and nonbasic variables is given as $x_N = (2 \ -1 \ 1)x_B - 1$. Thus the solution of problem 7.12 is the solution of the following minimization problem

$$\begin{array}{ll} \text{minimize} & h(y) = f\left(\left(y^T, (2y_1 - y_2 + y_3 - 1)^T\right)^T\right) \\ \text{subject to} & y \in \mathbb{R}^3. \end{array}$$

Example 7.10 *(Mad 1 in [61])*

$$\begin{array}{ll} \textit{minimize} & f(x) = \max_{i \le i \le 3} f_i(x) \\ \textit{subject to} & x_1 + x_2 - 0.5 \le 0 \end{array}$$

where $f_1(x) = x_1^2 + x_2^2 + x_1x_2 - 1, \quad f_2(x) = \sin(x_1), \quad f_3(x) = -\cos(x_2)$ *and* $x \in \mathbb{R}^2$.

We used a slack variable s into constraint, so that the constraint becomes $x_1 + x_2 - 0.5 - s = 0$, where $s \le 0$. Now we have, three variables, namely, x_1, x_2 and s, we divide the variables as $((x_B)^T, x_N) = (x_2, s, x_1)$. After some arrangement, we have the following unconstrained problem:

$$\begin{array}{ll} \text{minimize} & h(y) = \max_{i \le i \le 3} f_i\left(\left((0.5 - y_1 + y_2^2)^T, (y_1)^T\right)^T\right) \\ \text{subject to} & y \in \mathbb{R}^2. \end{array}$$

Example 7.11 *(Mad 2 in [61])*

$$\begin{array}{ll} \textit{minimize} & f(x) = \max_{i \le i \le 3} f_i(x) \\ \textit{subject to} & -3x_1 - x_2 - 2.5 \le 0 \end{array}$$

where $f_1(x) = x_1^2 + x_2^2 + x_1 x_2 - 1, \quad f_2(x) = \sin(x_1), \quad f_3(x) = -\cos(x_2)$ *and* $x \in \mathbb{R}^2$.

After some arrangment of the variables, we obtain the unconstrained problem

$$\begin{array}{ll} \text{minimize} & h(y) = \max_{i \le i \le 3} f_i\left(\left((-2.5 - y_1 - y_2^2)^T, y_1^T\right)^T\right) \\ \text{subject to} & y \in \mathbb{R}^2. \end{array}$$

The numerical results were obtained by applying all algorithms starting from 20 randomly generated points for each problem. Those results are given in Table 7.1, where the following notation is used:

- f_{ob}: the value of objective function f,
- n_f: number of objective function f evaluates,
- n_{sub}: number of subgradient evaluations respectively.

Table 7.1: Results of numerical experiments

Starting Points	Problem 1			Mad 1			Mad 2		
	f_{ob}	n_f	n_{sub}	f_{ob}	n_f	n_{sub}	f_{ob}	n_f	n_{sub}
1	0	139	86	-.38965952	267	109	-.33028514	127	80
2	0	152	89	-.38965952	195	122	-.33028514	123	75
3	0	132	76	-.38965952	276	115	-.33028514	140	85
4	0.1E-07	192	104	-.38965952	270	115	-.33028514	99	65
5	0.2E-07	187	94	-.38965952	243	106	-.33028514	112	72
6	0	156	98	-.38965952	280	120	-.33028514	120	76
7	0	121	73	-.38965952	373	136	-.33028514	107	66
8	0.1E-07	182	102	-.38965952	279	120	-.33028514	116	70
9	0.5E-07	187	88	-.38965952	250	102	-.33028514	110	69
10	0	117	80	-.38965952	347	133	-.33028514	116	74
11	0	149	88	-.38965952	231	108	-.33028514	125	71
12	0.4E-07	426	192	-.38965952	252	120	-.33028514	128	83
13	0	152	94	-.38965952	324	133	-.33028514	93	63
14	0.1E-07	147	90	-.38965952	249	113	-.33028514	121	74
15	0	152	94	-.38965952	345	141	-.33028514	120	73
16	0.1E-07	169	92	-.38965952	223	107	-.33028514	107	69
17	0.2E-07	101	100	-.38965952	242	115	-.33028514	127	80
18	0	101	68	-.38965952	243	110	-.33028514	131	81
19	0.1E-07	109	68	-.38965952	235	114	-.33028514	119	72
20	0.2E-07	211	104	-.38965952	292	135	-.33028514	130	80

The computational results show that the proposed method is not sensitive to the choice of the starting points. The numerical results are compared with those results in [13, 61]. The result of Example 7.12

in [13] is 0. The results in [61] for Example 7.10 is $-.3896592$ and for Example 7.11 is $-.33035714$. Example 7.12 and 7.10 are computed with high accuracy, in Example 7.11 the average error is $0.82E-05$. In Example 7.10, the proposed method used more function and subgradient evaluations than in the other examples.

CHAPTER 8

A GENERALIZED SUBGRADIENT METHOD WITH PIECEWISE LINEAR SUBPROBLEM

In this chapter, a new version of the quasisecant method for nonsmooth nonconvex optimization problem is developed. Quasisecants are overestimates to the objective function in some neighborhood of a given point. Subgradients are used to obtain quasisecants. We describe classes of nonsmooth functions, where quasisecants can be computed explicitly. It is shown that a descent direction with sufficient decrease must satisfy a set of linear inequalities. In the proposed algorithm, this set of linear inequalities is solved by applying the subgradient algorithm to minimize a piecewise linear function. We compare numerical results generated by the proposed algorithm and a subgradient method.

8.1 Introduction

Consider the following unconstrained minimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array} \tag{8.1}$$

where the objective function f is locally Lipschitz. Over the last four decades, subgradient [83], bundle [33, 36, 42, 47, 64, 65, 86], and the discrete gradient methods [5, 8, 14, 7, 13] have been proposed for solving this problem.

Among these methods, the subgradient method is the simplest one, although its convergence is proved only under convexity assumption on the function f ([17, 78, 83] for details). Better convergence results were obtained when the minimum value f^* of the objective function f is known. The aim of this chapter is to develop a method, which has simple and easy implementation and is applicable to wide range of nonsmooth optimization problems. We show that descent directions are the solutions of a system of linear inequalities. In the proposed algorithm, the solution of this system is reduced to convex problem taking a value of $f^* = 0$, when all inequalities are satisfied. We apply the subgradient method with known f^* to solve this problem. The convergence of the proposed algorithm is studied and results of numerical experiments are reported.

The structure of this chapter is as follows. We present an algorithm in order to find descent directions in Section 8.2. A description of the minimization algorithm is given in Section 8.3. We present the results of numerical experiments in Section 8.4. Section 8.5 concludes this chapter.

8.2 Computation of a Descent Direction

From now on, It is assumed that for any bounded subset $X \subset \mathbb{R}^n$ and any $h_0 > 0$, there exists a number $K > 0$ such that

$$\|v\| \leq K$$

for all $v \in QSec(x, h),\ x \in X$ and $h \in (0, h_0]$. Obviously, this assumption holds for all SR-quasisecants. Consider the following set for given $x \in \mathbb{R}^n$ and $h > 0$,:

$$W(x, h) = \text{cl co } QSec(x, h).$$

where cl co is the closed convex hull of a set. The set $W(x, h)$ is clearly compact and convex. It is shown in [10] that this set can be used to order to find descent directions of the objective function f in problem (8.1). However, the computation of the entire set $W(x, h)$ is not always possible. We propose an algorithm for computation of descent directions, and this algorithm uses only a few elements from $W(x, h)$. It is similar to that of proposed in [2], but the main step now consists in solving a linear system, much simpler than the system in the algorithm from [2]. Next, we describe an algorithm for finding search directions. Let numbers $c \in (0, 1)$ and $\delta > 0$ be given.

Algorithm 8.1 *Computation of the descent direction.*

Step 1. Chose any $g^1 \in S_1$, compute a quasisecant $v^1 = v(x, g^1, h)$. Set $V_1(x) = \{v^1\}$ and $k = 1$.

Step 2. We solve the following linear system of inequalities:

$$\langle v^i, g\rangle + \delta \leq 0;\ i = 1, \dots, k;\ g \in S_1. \tag{8.2}$$

Step 3. If the system (8.2) is not solvable, then terminates. Otherwise, compute a solution $\bar{g}$ of this system and set $g^{k+1} = \bar{g}$.

Step 4. If

$$f(x + hg^{k+1}) - f(x) \leq -ch\delta, \tag{8.3}$$

then stop.

Step 5. Compute a quasisecant $v^{k+1} = v(x, g^{k+1}, h)$ in the computed direction g^{k+1}, construct the set $V_{k+1}(x) = \text{co}\left\{V_k(x) \bigcup \{v^{k+1}\}\right\}$, set $k \leftarrow k + 1$ and go to Step 2.

One can see that Algorithm 8.1 computes quasisecants step by step (in Steps 1 and 5) until one of the conditions satisfies: either system (8.2) is not solvable or inequality (8.3) is true. Condition (8.3) means that the descent direction has been found. The situation when system (8.2) is not solvable is considered in Proposition 8.2 below. An algorithm for solving the system (8.2) is proposed in Subsection 8.2.1.

Proposition 8.2 *If system (8.2) is not solvable, then*

$$\min_{v \in V_k(x)} \|v\|_\infty < \delta. \tag{8.4}$$

Proof: Let $\tilde{v}$ denote a solution of the following problem:

$$\min \frac{1}{2}\|v\|_2^2 \quad \text{subject to} \quad v \in V_k(x).$$

If $\tilde{v} = 0$, then the proof is straightforward. Thus, we assume that $\tilde{v} \neq 0$. It follows from the necessary condition for a minimum that

$$\langle \tilde{v}, v - \tilde{v} \rangle \geq 0, \quad \forall v \in V_k(x),$$

i.e.,

$$\|\tilde{v}\|_2^2 \leq \langle \tilde{v}, v \rangle, \quad \forall v \in V_k(x).$$

Then,

$$\|\tilde{v}\|_\infty^2 \leq \langle \tilde{v}, v \rangle, \quad \forall v \in V_k(x). \tag{8.5}$$

Since the system (8.2) is not solvable

$$\max_{i=1,\ldots,k} \langle v^i, g \rangle > -\delta, \quad \forall g \in S_1.$$

Then, for $g^\infty = -\|\tilde{v}\|_\infty^{-1}\tilde{v}$ there exists v^i ($i \in \{1, \ldots, k\}$) such that

$$\langle v^i, \tilde{v} \rangle < \delta \|\tilde{v}\|_\infty.$$

Then, using (8.5), we complete the proof. $\triangle$

Remark 8.3 It follows from Proposition 8.2 that if system (8.2) is not solvable, then the point $x \in \mathbb{R}^n$ can be considered as an approximate solution.

Proposition 8.4 *Assume that a function f is locally Lipschitz defined on $\mathbb{R}^n$. The Algorithm 8.1 terminates after finite number of iterations.*

Proof: If both stoping criteria of the Algorithm 8.1 are not held, then the computed quasisecant $v^{k+1} \notin V_k(x)$, in other words the set $V_k(x)$ can be improved by adding the computed quasisecant v^{k+1}. Indeed, in this case

$$f(x + hg^{k+1}) - f(x) > -ch\delta.$$

It follows from the definition of the quasisecants that

$$f(x + hg^{k+1}) - f(x) \leq h\langle v^{k+1}, g^{k+1} \rangle,$$

which means that

$$\langle v^{k+1}, g^{k+1} \rangle > -c\delta. \tag{8.6}$$

We assume that $v^{k+1} \in V_k(x)$. Since $g^{k+1} \in S_1$ is a solution of system (8.2), we get

$$\langle v^i, g^{k+1} \rangle + \delta \leq 0 \ (i = 1, \ldots, k)$$

we have

$$\langle v^{k+1}, g^{k+1} \rangle \leq -\delta,$$

which contradicts (8.6). Therefore, $v^{k+1} \notin V_k(x)$.

Now, we will show that Algorithm 8.1 terminates. Assume that Algorithm 8.1 generates an infinite sequence $\{g^k\}$ of directions $g^k \in S_1$. It follows from (8.6) that

$$\langle v^k, g^k \rangle > -c\delta, \quad \forall\, k = 2, 3, \ldots. \tag{8.7}$$

This implies that for any $k \in \{2, 3, \dots\}$ the direction g^k does not satisfy the following system:

$$\langle v^t, g \rangle + \delta \le 0, \ t = 1, \dots, i, \ i \ge k.$$

Since the set $W(x, h)$ is compact, there exists a number $\overline{C} > 0$ such that $\|v\|_2 \le \overline{C}$ for all $v \in V_k(x)$. The direction g^{k+1} is a solution of the system

$$\langle v^i, g \rangle + \delta \le 0 \ i \ (1, \dots, k).$$

However, directions $g^j, \ j = 2, \dots, k$ are not solutions of this system. Then,

$$\|g^{k+1} - g^j\|_\infty > \frac{(1-c)\delta}{\overline{C}\sqrt{n}}, \ \ \forall \ j = 2, \dots, k. \tag{8.8}$$

Indeed, if there exists $j \in \{2, \dots, k\}$ such that

$$\|g^{k+1} - g^j\|_\infty \le \frac{(1-c)\delta}{\overline{C}\sqrt{n}},$$

then we have

$$\|g^{k+1} - g^j\|_2 \le \frac{(1-c)\delta}{\overline{C}}$$

and

$$\left| \langle v^j, g^{k+1} \rangle - \langle v^j, g^j \rangle \right| \le (1-c)\delta.$$

Hence,

$$\langle v^j, g^j \rangle \le \langle v^j, g^{k+1} \rangle + (1-c)\delta \le -c\delta,$$

which contradicts (8.7). Inequality (8.8) can be rewritten as follows:

$$\min_{j=2,\dots,k} \|g^{k+1} - g^j\|_\infty > \frac{(1-c)\delta}{\overline{C}\sqrt{n}}.$$

Thus Algorithm 8.1 generates a sequence $\{g^k\}$ of directions $g^k \in S_1$ such that the distance between g^k and the set of all previous directions is bounded below. Since the set S_1 is bounded the number of such directions is finite. △

Definition 8.5 *A point $x \in \mathbb{R}^n$ is called an* (h, δ)-stationary point *if*

$$\min_{v \in W(x,h)} \|v\|_\infty \le \delta.$$

One can see that after finitely many iterations, Algorithm 8.1 either finds that the point x is the (h, δ)-stationary or it finds the direction of sufficient decrease at this point satisfying the inequality (8.3).

8.2.1 Solving System (8.2)

The problem of finding of descent directions in Algorithm 8.1 is reduced to the solution of a system of linear inequalities (8.2). Different algorithms for solving the system linear inequalities can be found, for example, in [40]. Here we apply the subgradient method to solve such systems.

Letting

$$\varphi(g) = \max\left\{0, \ \langle v^i, g \rangle + \delta, \ i = 1, \dots, m\right\},$$

one can show that finding a solution of this system can be reduced to the following minimization problem:

$$\begin{aligned} &\text{minimize } \varphi(g) \\ &\text{subject to } g \in B_1. \end{aligned} \tag{8.9}$$

Here $B_1 = \{g \in \mathbb{R}^n \mid \|g\|_\infty \le 1\}$. The function φ is convex piecewise linear and it is Lipschitzian. Let $\bar{g} \in B_1$ be a solution of problem (8.9). If $\varphi(\bar{g}) = 0$, then $\bar{g}$ is a solution to the system (8.2). If $\varphi(\bar{g}) > 0$, then the system (8.2) is not solvable. In order to solve problem (8.9), we reduce it to a unconstrained minimization of a convex piecewise linear function.

The Lipschitz constant K_2 of the function φ in Euclidean norm L_2 is

$$K_2 = \max_{i=1,\dots,m} \|v^i\|_2$$

and its Lipschitz constant K_∞ in L_∞ norm is

$$K_\infty = K_2 \sqrt{n}.$$

For a given point $y \in \mathbb{R}^n$ and a given set $G \subset \mathbb{R}^n$, the L_∞-distance between y and G is defined as follows

$$d_\infty^G(y) := \inf \{\|y - x\|_\infty \mid x \in G\}.$$

If $G = B_1$, then

$$d_\infty^{B_1}(y) = \max \{0, y_i - 1, -y_i - 1 \; (i = 1, \dots, n)\}.$$

The following lemma can be found, for example, in [64]. Let d^G be a distance function based on a given norm.

Lemma 8.6 *Let f be a Lipschitz continuous function with constant $K > 0$ on a set $S \subset \mathbb{R}^n$. Let $x \in G \subset S$ and suppose that f attains a minimum over G at x. Then, for any $\hat{K} \ge K$ the function $\psi(y) := f(y) + \hat{K} d^G(y)$ attains a minimum over S at x. If $\hat{K} > K$ and G is closed, then every minimizer of ψ over S lies in G.*

Let

$$\Phi(g) := \varphi(g) + K_\infty d_\infty^G(g).$$

It follows from Lemma 8.6 that problem (8.9) can be reduced to the following unconstrained minimization problem:

$$\begin{aligned} &\text{minimize } \Phi(g) \\ &\text{subject to } g \in \mathbb{R}^n. \end{aligned} \tag{8.10}$$

The function Φ is convex piecewise linear.

Thus the problem of finding $g \in \mathbb{R}^n$ satisfying the system (8.2) is reduced to solving problem (8.10). This approach has one clear advantage. Since one repeatedly solves the system (8.2) adding one inequality, every time until we find a descent direction, we can reuse the solution to the previous system each time. This solution violates only the new inequality which shows that it is close to the set of solutions of the new system.

We suggest to apply the subgradient method with known minimum objective value to solve problem (8.10). Let $g^0 \in \mathbb{R}^n$ be a starting point. Then, the subgradient method proceeds as follows (see [78]):

$$g^{k+1} = g^k - \alpha \frac{\Phi(g^k) - \bar{\Phi}}{\|w^k\|_2^2} w^k \; (k = 0, 1, \dots),$$

where $\alpha \in (0, 2)$, $w^k \in \partial\Phi(g^k)$ is a subgradient and $\bar{\Phi}$ is an underestimate for minimum value. Here, $\bar{\Phi} = 0$. Although this algorithm provides only an approximate solution, it is very simple and requires only one subgradient at each iteration.

Remark 8.7 The choice of δ in Algorithm 8.1 is very important. Large values of δ may lead to the system of inequalities (8.2) which is not solvable although the directions of significant decrease may exist. For small values of δ, the algorithm may generate directions of an insignificant decrease, although the directions of much better decrease may exist. We propose the following approach to avoid this problem. Let be given $\Delta > 1$; let also $\eta > 0$ be a tolerance (say $\eta = 10^{-8}$). Then, we solve problem (8.10) for $\delta = \Delta$. Let $\bar{g} \in B_1$ be its solution and $\bar{\delta} = \Phi(\bar{g})$. If $\bar{\delta} \geq \Delta$, then we accept that system (8.2) is not solvable for any $\delta > 0$. Otherwise, we define $\delta_0 := \Delta - \bar{\delta}$. If $\delta_0 \leq \eta$, then the system (8.2) is not solvable for any $\delta > \eta$ and the point x is (h, η)-stationary point. If $\delta_0 > \eta$, then the system (8.2) has a solution for $\delta = \delta_0$. We take $\alpha := 1/\Delta$ in the subgradient method.

8.3 A Minimization Algorithm

In this section, we describe minimization algorithms for solving problem (8.1). First we will describe an algorithm for finding (h, δ)-stationary points of the objective function f.

Let $h > 0$, $\delta > 0$, $c_1 \in (0, 1)$, $c_2 \in (0, c_1]$ be given numbers.

Algorithm 8.8 *The quasisecant method for finding a (h, δ)-stationary points.*

Step 1. Start with any point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Apply Algorithm 8.1 setting $x = x^k$. This algorithm terminates after finite number $l > 0$ of iterations. s a result we get the system:

$$\langle v^i, g \rangle + \delta \leq 0 \ (i = 1, \dots, l), \ g \in S_1. \tag{8.11}$$

Step 3. If this system is not solvable then stop, x^k is the (h, δ)-stationary point. Otherwise, we get the direction $g^k \in S_1$ which is a solution to this system and

$$f(x^k + hg^k) - f(x^k) \leq -c_1 h\delta. \tag{8.12}$$

Step 4. Compute $x^{k+1} = x^k + \sigma_k g^k$, where σ_k is defined as follows:

$$\sigma_k = \operatorname{argmax} \left\{ \sigma \geq 0 \mid \ f(x^k + \sigma g^k) - f(x^k) \leq -c_2 \sigma\delta \right\}. \tag{8.13}$$

Set $k \leftarrow k + 1$ and go to Step 2.

Theorem 8.9 *Let us assume that the function f is bounded from below:*

$$f_* = \inf \{f(x) \mid x \in \mathbb{R}^n\} > -\infty. \tag{8.14}$$

Then, Algorithm 8.8 terminates after finitely many iterations $M > 0$ and produces a (h, δ)-stationary point x^M, where

$$M \leq M_0 \equiv \left\lfloor \frac{f(x^0) - f_*}{c_2 h\delta} \right\rfloor + 1.$$

Here, $\lfloor u \rfloor$ shows the integer part of the number $u > 0$.

Proof: We assume the contrary. Then, the sequence $\{x^k\}$ is infinite and points x^k are not (h,δ)-stationary points. This means that

$$\min\{\|v\|_\infty \mid v \in W(x^k,h)\} > \delta, \quad \forall k = 1,2,\ldots.$$

Therefore, Algorithm 8.1 will find descent directions and the inequality (8.11) will be satisfied at each iteration k. Since $c_2 \in (0,c_1]$, it follows from (8.12) that $\sigma_k \geq h$. Therefore, we have

$$\begin{aligned} f(x^{k+1}) - f(x^k) &< -c_2\sigma_k\|v^k\|_\infty \\ &\leq -c_2h\|v^k\|_\infty. \end{aligned}$$

Since $\|v^k\|_\infty > \delta$ for all $k \geq 0$, we get

$$f(x^{k+1}) - f(x^k) \leq -c_2h\delta,$$

which implies

$$f(x^{k+1}) \leq f(x^0) - (k+1)c_2h\delta$$

and, therefore, $f(x^k) \to -\infty$ as $k \to +\infty$, which contradicts condition (8.14). It is obvious that the upper bound for the number of iterations M necessary to find the (h,δ)-stationary point is M_0. △

Remark 8.10 Because of the fact that $c_2 \leq c_1$ and $\sigma_k \geq h$, $h > 0$ is a lower bound for σ_k. This allows us to estimate σ_k by using the following rule:
σ_k is defined as the largest $\theta_l = 2^l h \quad (l \in \mathbb{N})$, satisfying the inequality in Equation 8.13.

Algorithm 8.8 can be applied to compute stationary points of the function f. Let $\{h_k\}$ be a sequence such that $h_{k+1} = \gamma h_k,\ \gamma \in (0,1),\ h_0 > 0$ and $\varepsilon, \eta > 0$ be given tolerances.

Algorithm 8.11 *The quasisecant method with piecewise linear subproblem.*

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$, and set $k = 0$.

Step 2. If $h_k < \varepsilon$, then stop.

Step 3. Apply Algorithm 8.8 starting from the point x^k for $h = h_k$ and $\delta = \eta$. This algorithm terminates after finitely many iterations $M > 0$, and as a result, it computes an (h_k,η)-stationary point x^{k+1}.

Step 4. Set $k \leftarrow k+1$ and go to Step 2.

Remark 8.12 Following Remark 8.7 one can apply Algorithm 8.8 in Step 3 as follows. We take a sufficiently large $\Delta > 0$ and apply Algorithm 8.1 in Step 2 of Algorithm 8.8 with $\delta = \Delta$ and then compute $\delta_0 = \Delta - \bar{\delta}$ (see Remark 8.7). If $\delta_0 < \eta$ then (h_k,η)-stationary point has been computed. Such an approach will accelerate the convergence of Algorithm 8.11.

For the point $x^0 \in \mathbb{R}^n$, consider the set

$$\mathcal{L}(x^0) = \left\{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\right\}.$$

Theorem 8.13 *We assume that the function f is locally Lipschitz continuous, the set $W(x,h)$ is constructed using SR-quasisecants, condition (2.11) is satisfied and the set $\mathcal{L}(x^0)$ is bounded for starting points $x^0 \in \mathbb{R}^n$. Then, every accumulation point of the sequence $\{x^k\}$ belongs to the set $X^0 = \{x \in \mathbb{R}^n \mid 0 \in \partial f(x)\}$.*

Proof: Since the function f is locally Lipschitz and the set $\mathcal{L}(x^0)$ is bounded, $f_* > -\infty$. Therefore, conditions of Theorem 8.9 are satisfied, and Algorithm 8.8 generates a sequence of (h_k, η)-stationary points ($k \geq 0$) after the finite number of steps. Since for any $k > 0$, the point x^{k+1} is an (h_k, η)-stationary, it follows from the definition of the (h_k, η)-stationary points that

$$\min\left\{\|v\|_\infty \mid v \in W(x^{k+1}, h_k)\right\} \leq \eta. \tag{8.15}$$

It is obvious that $x^k \in \mathcal{L}(x^0)$ for all $k \geq 0$. The boundedness of the set $\mathcal{L}(x^0)$ implies that the sequence $\{x^k\}$ has at least one accumulation point. Let x^* be an accumulation point and $x^{k_i} \to x^*$ as $i \to +\infty$. The inequality in (8.15) implies that

$$\min\left\{\|v\|_\infty \mid v \in W(x^{k_i}, h_{k_i-1})\right\} \leq \eta. \tag{8.16}$$

The mapping $QSec(\cdot,\cdot)$ satisfies the condition (2.11), therefore, at the point x^* for any $\mu > 0$ there exists a number $\nu > 0$ such that

$$W(y, h) \subset \partial f(x^*) + B_\mu \tag{8.17}$$

for all $y \in B_\nu(x^*)$ and $h \in (0, \nu)$. Since the sequence $\{x^{k_i}\}$ converges to x^*, there exists $i_0 > 0$ such that $x^{k_i} \in B_\nu(x^*)$ for all $i \geq i_0$. On the other hand, since $\delta_k, h_k \to +0$ as $k \to +\infty$, there exists $k_0 > 0$ such that $h_k < \nu$ for all $k > k_0$. Then there exists $i_1 \geq i_0$ such that $k_i \geq k_0 + 1$ for all $i \geq i_1$. Thus, it follows from (8.16) and (8.17) that

$$\min\{\|v\| \mid \ v \in \partial f(x^*)\} \leq \mu + \nu.$$

Since $\mu > 0$ and $\nu > 0$ are chosen arbitrarily and the mapping $x \mapsto \partial f(x)$ is upper semicontinuous, $0 \in \partial f(x^*)$. △

8.4 Numerical Experiments

Numerical results was obtained by applying the proposed method to some academic test problems with nonsmooth objective functions. These numerical results were used to verify the efficiency of the proposed method by comparing the numerical results of a subgradient method. The test problems are taken from [61]. Brief description of test problems can be found in table 8.1, where the following notation is used:

- n: number of variables,
- f_{opt}: optimal value.

In Algorithm 8.11, parameters and tolerances were chosen as follows: $\Delta = 10^4$, $c_1 = 0.2$, $c_2 = 0.05$, $\gamma = 0.5$, $\eta = 10^{-8}$, $\varepsilon = 10^{-10}$. We use the following stopping criteria in the subgradient method for solving problem (8.10). The algorithm stops if:

1. the number of function evaluations is more than 10000, or
2. it cannot decrease the value objective function $\Phi(g)$ in 1000 successive iterations.

We compare the proposed algorithm with the subgradient method (see [83]). Let $x^0 \in \mathbb{R}^n$ be a starting point. Then the subgradient method proceeds as follows.

$$x^{k+1} = x^k - \alpha_k v^k, \tag{8.18}$$

Table 8.1: The brief description of test problems

Problem	n	f_{opt}	Problem	n	f_{opt}
Crescent	2	0	Shor	5	22.600162
CB2	2	1.9522245	El-Attar	6	0.5598131
CB3	2	2	Gill	10	9.7857721
DEM	2	-3	Steiner 2	12	16.703838
QL	2	7.2	Maxq	20	0
LQ	2	-1.4142136	Maxl	20	0
Mifflin 1	2	-1	Goffin	50	0
Mifflin 2	2	-1	MXHILB	50	0
Wolfe	2	-8	L1HILB	50	0
Rosen-Suzuki	4	-44	Shell Dual	15	32.348679

where $v^k \in \partial f(x^k)$ is any subgradient and $\alpha_k > 0$ is a step-length. Convergence of the subgradient method was proved only for convex functions [83]. However, we apply this algorithm also to nonconvex problems. We use the following update for the step-length α_k. We take $\alpha_k = 1/k$, however, after each 25000 iterations we update it. Let p_k is the largest integer, smaller than or equal to $k/25000$. Then

$$\alpha_k = \frac{1}{k - 25000p_k}.$$

Without this update of α_k, the convergence of the subgradient method is extremely poor, especially, for nonconvex functions. We use the following two stopping criteria in the subgradient method. First, the number of function evaluations is restricted by 2×10^5. Second, the algorithm stops if it cannot decrease the value objective function in 1000 successive iterations.

Numerical experiments were carried out by using a computer, whose configuration is Intel Pentium 4 processor (1.83 GHz) and 1GB of RAM. For the calculation of both methods, the same 20 random generated starting points for each problem are used.

When comparing the performance of the methods, two indicators are used, namely

- n_b: the number of successful solved problems according to the following relative error considering the best known solution reported in [61],
- n_s: the number of successful solved problems according to the following relative error considering the best found solution by these two algorithms.

We assume that f_{opt} and $\bar{f}$ are the values of the objective function at the best-known solution in [61] and at the best found solution by these two algorithms, respectively. Then, If the following inequality holds, then an algorithm solves successfully the problem with respect to a tolerance $\varepsilon > 0$:

$$\frac{f_* - f_0}{1 + |f_*|} \leq \varepsilon,$$

where f_* is equal either to f_{opt} (for n_b) or to $\bar{f}$ (for n_s) and f_0 is the optimal value of the objective function found by an algorithm. In numerical calculation, we use the tolerance $\varepsilon = 10^{-4}$.

Results of numerical experiments are presented in Tables 8.2 - 8.3. In Table 8.2, we report the relative error (E) for the average objective function value (f_{av}) over 20 runs of the algorithms as well as the numbers n_b and n_s for each problem. We compute the relative error E as follows:

$$E = \frac{f_{av} - f_{opt}}{1 + |f_{opt}|}.$$

If $E < 10^{-5}$, then in the table we put $E = 0$.

Table 8.2: Results of numerical experiments: obtained solutions

	The proposed algorithm			Subgradient method		
Problem	E	n_b	n_s	E	n_b	n_s
Crescent	0	20	20	0	20	20
CB2	0	20	20	0	20	20
CB3	0	20	20	0	20	20
DEM	0	20	20	0	20	20
QL	0	20	20	0	20	20
LQ	0.00003	18	20	0	20	20
Mifflin 1	0	20	20	0	20	20
Mifflin 2	0	20	20	0	20	20
Wolfe	0	20	20	0	20	20
Rosen-Suzuki	0.00001	20	20	0	20	20
Shor	0.00005	20	20	0.00002	20	20
El-Attar	0.38987	13	19	2.26980	1	2
Gill	0.00006	17	20	0.01117	0	1
Steiner 2	0	20	20	0.01366	0	0
MAXQ	0	20	20	107.87845	0	0
MAXL	0	20	20	10.33056	0	0
Goffin	0	20	20	826.90962	0	0
MXHILB	0.02376	0	17	0.10274	0	3
L1HILB	0.03189	0	20	0.32035	0	0
Shell Dual	161.88049	0	5	81.62459	0	15

The results presented in Table 8.2 show that the proposed algorithm outperforms the subgradient method. The difference between these two algorithms becomes larger as the number of variables increases. Results for the relative error (E) and also for n_b and n_s for the El-Attar, Gill, Steiner 2, MAXQ, MAXL, Goffin, MXHILB and L1HILB the test problem confirm this claim. The results for the El-Attar, Gill, Steiner 2 the test problems show that the proposed algorithm is more efficient than the subgradient method for solving nonconvex nonsmooth problems. The subgradient method produces better result for only Dual Shell problem.

Table 8.3 presents the average number of iterations (n_i), the objective function and subgradient evaluations (n_f and n_{sub}, respectively) and the average CPU time over 20 runs of algorithms. Since for the subgradient method the number of iterations, the objective function and subgradient evaluations are the same we present only one of them.

One can see from results presented in Table 8.3 that the proposed algorithm requires significantly less number of the objective function and subgradient evaluations. This means that if the objective function

Table 8.3: Results for the number of function and subgradient evaluations, CPU time

	The proposed method				Subgradient method	
Problem	n_i	n_f	n_{sub}	CPU	$n_i(n_f, n_{sub})$	CPU
Crescent	60	200	113	0.01	31748	0.01
CB2	97	316	206	0.02	26387	0.01
CB3	88	298	239	0.03	16123	0.01
DEM	92	312	244	0.03	33507	0.01
QL	93	291	187	0.02	69682	0.01
LQ	81	202	159	0.01	19021	0.00
Mifflin 1	90	285	180	0.02	33933	0.01
Mifflin 2	91	267	179	0.02	21744	0.00
Wolfe	85	231	174	0.02	16631	0.00
Rosen-Suzuki	108	433	292	0.04	60989	0.01
Shor	108	517	387	0.07	44675	0.01
El-Attar	202	1261	867	0.27	162517	7.81
Gill	135	654	388	0.19	142834	33.28
Steiner 2	225	2064	1707	0.82	175265	0.41
MAXQ	269	2575	389	0.12	200000	0.11
MAXL	129	1003	832	0.63	200000	0.10
Goffin	258	8666	8118	29.36	200000	0.29
MXHILB	161	815	361	0.34	191376	8.53
L1HILB	385	2331	955	0.91	189336	15.83
Shell Dual	221	1628	329	0.06	200000	0.24

is complex enough, then the proposed algorithm uses significantly less CPU time than the subgradient method. Results for the El-Attar, Gill, MXHILB, L1HILB and Shell Dual the test problems confirm this claim.

8.5 Conclusion

In this chapter, we have proposed a generalized subgradient algorithm. In this algorithm, a descent direction is found solving the system of linear inequalities. In order to solve this system we apply the subgradient method with known minimum value of the objective function. The proposed algorithm is easy to implement. Results of numerical experiments show that the proposed algorithm outperforms the subgradient method for the most of the test problems used in this chapter. The new algorithm requires significantly less function and subgradient evaluations than the subgradient method. The proposed algorithm allows one to significantly reduce CPU time by the subgradient method on problems with complex objective functions.

CONCLUSION AND OUTLOOK

The aim of this thesis is to develop methods for nonsmooth optimization problems. The nonsmooth optimization theory can be split into two parts as convex problems and nonconvex problems and we aimed to develop the methods for both types of optimization problems. In this thesis, for the convex problems, we proposed three methods, while for nonconvex problem just one method was developed in order to seek the minimum of objective functions, which are locally Lipschitzian functions. Additionally, an adaptation of one of these three method for convex problems was presented.

For convex functions, three methods have been developed, namely Truncated Codifferential Methods (TCM), Truncated Codifferential Method with Memory (TCMWM) and Aggregate Codifferential Method (ACM). In TCM, only a few elements of the codifferential are used to calculate search directions. In order to reduce the number of codifferential evaluations, TCMWM has been developed using the codifferential calculated in previous iterations to calculate search directions. In both the methods TCM and TCMWM, search directions are computed solving a quadratic subproblem. The size of this problem can increase significantly as the number of variables increases. In order to overcome this problem, ACM has been developed. In ACM, the number of elements of the codifferential used to find search directions is restricted, which allows us to apply ACM to large scale optimization porblems. The theoretical proofs of convergence for all proposed methods were given and their validation were tested on a wide range of well-known test problems. After that, TCM was adapted for linearly constrained nonsmooth convex optimization problems. It can be an example of how our methods (TCM, TCMWM and ACM) can be used for constrained problems. In the following part of the thesis, a generalization of subgradient method (GSM) was given for locally Lipschitz continues functions. To find search direction, a linear inequality system has been solved, which is an important part of GSM. The convergency of GSM was proved and it was tested on general nonsmooth unconstrained optimization problems.

Aforementioned three methods for convex problems can be briefly described as follows. The first method, TCM, is important in terms of using the codifferential concept, due to rare usage of codifferential in the literature. Actually, a codifferential has good differential properties in order to develop optimization methods. Thus, TCM can be refined for the use on the other types of nonsmooth optimization problems. Regarding the description of the methods, in TCM, at each iteration, only a few elements from the hypodifferential of the objective function were used to compute descent directions. It was proved that the proposed method converged to minimizers of a convex function. For the second method, the aim was to reduce the number of function evaluations, and, especially the gradient evaluations. Thus, we used some calculated hypodifferential from previous iteration, however, it was difficult to decide which hypodifferentials were usable and how we could choose them. How these decisions were made clearly explained, and it was proved that this selection allowed us to find descent directions. The last method, ACM for nonsmooth convex unconstrained problems, was developed to reduce the size of the aforementioned quadratic subproblem. According to this purpose, we used aggregate information via the fixed number hypodifferentials, where the number arranges the simplicity of ACM. Such an approach allows one to significantly reduce the complexity of codifferential methods.

The proposed method for linearly constrained optimization problems is just the converted version of

TCM for convex unconstrained optimization problems. In this thesis, it was explained how we convert linearly constrained optimization problems to unconstrained problems, and that convexity assumption is preserved.

A new version of the quasisecant method for nonsmooth nonconvex optimization problem was developed. Descent directions were computed by solving a subproblem, which is the system of linear inequalities. For the proposed method, implementation was easy. The numerical experiments demonstrated that a generalized subgradient method significantly reduced the cpu time compared to the subgradient method on general nonsmooth optimization problems.

Future Works

For further investigations, we can suggest several possible projects. First, codifferential methods can be generalized to nonsmooth nonconvex optimization problems, such as locally Lipschitz functions, minmax problems, differences of two convex functions (DC) and differences of two polyhedral functions (DP). Generalizability of codifferential methods can be reached due to codifferential has good differential properties and it can be explicitly given for the set which consists of the codifferential for that class of functions. As the Second suggestion, codifferential methods and the subgradient method can be improved for nonsmooth constrained optimization problems by using exact penalty functions or slack variables, as shown in Section 7. Finally, as the third suggestion, the subgradient method can be improved in order to make the proposed algorithm more efficient, and to provide better algorithms for solving subproblems. This as well as the comparison of the proposed algorithm with the bundle method will be a topic for future research.

REFERENCES

[1] K. M. Anstreicher and L. A. Wolsey. Two 'well-known' properties of subgradient optimization. *Math. Program.*, 120(1, Ser. B):213–220, 2009.

[2] A. Bagirov and A. N. Ganjehlou. An approximate subgradient algorithm for unconstrained nonsmooth, nonconvex optimization. *Math. Methods Oper. Res.*, 67(2):187–206, 2008.

[3] A. Bagirov and J. Ugon. Codifferential method for minimizing nonsmooth dc functions. *Journal of Global Optimization*, 50(1):3–22, May 2011.

[4] A. M. Bagirov. A method for the approximation of a subdifferential. *Zh. Vychisl. Mat. i Mat. Fiz.*, 32(4):652–658, 1992.

[5] A. M. Bagirov. Minimization methods for one class of nonsmooth functions and calculation of semi-equilibrium prices. In *Progress in optimization*, volume 30 of *Appl. Optim.*, pages 147–175. Kluwer Acad. Publ., Dordrecht, 1999.

[6] A. M. Bagirov. Numerical methods for minimizing quasidifferentiable functions: a survey and comparison. In *Quasidifferentiability and related topics*, volume 43 of *Nonconvex Optim. Appl.*, pages 33–71. Kluwer Acad. Publ., Dordrecht, 2000.

[7] A. M. Bagirov. A method for minimization of quasidifferentiable functions. *Optim. Methods Softw.*, 17(1):31–60, 2002.

[8] A. M. Bagirov. Continuous subdifferential approximations and their applications. *J. Math. Sci. (N. Y.)*, 115(5):2567–2609, 2003. Optimization and related topics, 2.

[9] A. M. Bagirov and A. N. Ganjehlou. A secant method for nonsmooth optimization. *Frontiers Science Series*, 49:243, 2007.

[10] A. M. Bagirov and A. N. Ganjehlou. A quasisecant method for minimizing nonsmooth functions. *Optim. Methods Softw.*, 25(1):3–18, 2010.

[11] A. M. Bagirov, A. N. Ganjehlou, H. Tor, and J. Ugon. A generalized subgradient method with piecewise linear subproblem. *Dyn. Contin. Discrete Impuls. Syst. Ser. B Appl. Algorithms*, 17(5):621–638, 2010.

[12] A. M. Bagirov, A. N. Ganjehlou, J. Ugon, and A. H. Tor. Truncated codifferential method for nonsmooth convex optimization. *Pac. J. Optim.*, 6(3):483–496, 2010.

[13] A. M. Bagirov, M. Ghosh, and D. Webb. A derivative-free method for linearly constrained nonsmooth optimization. *J. Ind. Manag. Optim.*, 2(3):319–338, 2006.

[14] A. M. Bagirov, B. Karasözen, and M. Sezer. Discrete gradient method: derivative-free method for nonsmooth optimization. *J. Optim. Theory Appl.*, 137(2):317–334, 2008.

[15] C. Beltran and F. J. Heredia. An effective line search for the subgradient method. *J. Optim. Theory Appl.*, 125(1):1–18, 2005.

[16] A. Ben-Tal and M. Teboulle. A smoothing technique for nondifferentiable optimization problems. In *Optimization (Varetz, 1988)*, volume 1405 of *Lecture Notes in Math.*, pages 1–11. Springer, Berlin, 1989.

[17] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999.

[18] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. A family of variable metric proximal methods. *Math. Programming*, 68(1, Ser. A):15–47, 1995.

[19] J. V. Burke, D. Henrion, A. S. Lewis, and M. L. Overton. Stabilization via nonsmooth, nonconvex optimization. *IEEE Trans. Automat. Control*, 51(11):1760–1769, 2006.

[20] J. V. Burke, A. S. Lewis, and M. L. Overton. Approximating subdifferentials by random sampling of gradients. *Math. Oper. Res.*, 27(3):567–584, 2002.

[21] J. V. Burke, A. S. Lewis, and M. L. Overton. Two numerical methods for optimizing matrix stability. *Linear Algebra Appl.*, 351/352:117–145, 2002. Fourth special issue on linear systems and control.

[22] J. V. Burke, A. S. Lewis, and M. L. Overton. Pseudospectral components and the distance to uncontrollability. *SIAM J. Matrix Anal. Appl.*, 26(2):350–361 (electronic), 2004/05.

[23] J. V. Burke, A. S. Lewis, and M. L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM J. Optim.*, 15(3):751–779, 2005.

[24] F. H. Clarke. *Optimization and nonsmooth analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. John Wiley & Sons Inc., New York, 1983. A Wiley-Interscience Publication.

[25] A. R. Conn and P. Toint. An Algorithm Using Quadratic Interpolation for Unconstrained Derivative Free Optimization. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications*, pages 27–47. Kluwer Academic/Plenum Publishers, New York, 1996.

[26] V. F. Demyanov, A. M. Bagirov, and A. M. Rubinov. A method of truncated codifferential with application to some problems of cluster analysis. *J. Global Optim.*, 23(1):63–80, 2002.

[27] V. F. Demyanov and A. M. Rubinov. *Constructive nonsmooth analysis*, volume 7 of *Approximation & Optimization*. Peter Lang, Frankfurt am Main, 1995.

[28] V.F. Demyanov and L.C.W. Dixon. *Quasidifferential calculus*. Amsterdam : North-Holland, 1986. "A publication of the Mathematical Programming Society.".

[29] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002.

[30] Yu. M. Ermoliev. Methods of nondifferentiable and stochastic optimization and their applications. In *Progress in nondifferentiable optimization*, volume 8 of *IIASA Collaborative Proc. Ser. CP-82*, pages 5–27. Internat. Inst. Appl. Systems Anal., Laxenburg, 1982.

[31] R. Fletcher. *Practical methods of optimization*. A Wiley-Interscience Publication. John Wiley & Sons Ltd., Chichester, second edition, 1987.

[32] A. Frangioni. Solving semidefinite quadratic problems within nonsmooth optimization algorithms. *Comput. Oper. Res.*, 23(11):1099–1118, 1996.

[33] A. Frangioni. Generalized bundle methods. *SIAM J. Optim.*, 13(1):117–156 (electronic), 2002.

[34] A. Fuduli, M. Gaudioso, and G. Giallombardo. Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *SIAM J. Optim.*, 14(3):743–756 (electronic), 2003.

[35] M. Fukushima and L. Qi. A globally and superlinearly convergent algorithm for nonsmooth convex minimization. *SIAM J. Optim.*, 6(4):1106–1120, 1996.

[36] M. Gaudioso and M. F. Monaco. A bundle type approach to the unconstrained minimization of convex nonsmooth functions. *Math. Programming*, 23(2):216–226, 1982.

[37] M. Haarala, K. Miettinen, and M. M. Mäkelä. New limited memory bundle method for large-scale nonsmooth optimization. *Optim. Methods Softw.*, 19(6):673–692, 2004.

[38] N. Haarala, K. Miettinen, and M. M. Mäkelä. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Math. Program.*, 109(1, Ser. A):181–205, 2007.

[39] W. Hare and C. Sagastizábal. Computing proximal points of nonconvex functions. *Math. Program.*, 116(1-2, Ser. B):221–258, 2009.

[40] K. L. Hiebert. Solving systems of linear equations and inequalities. *SIAM Journal on Numerical Analysis*, 17(3):pp. 447–464, 1980.

[41] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II*, volume 306 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1993. Advanced theory and bundle methods.

[42] J.B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms*, volume 1 and 2. Springer-Verlag, Heidelberg, 1993.

[43] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. I*, volume 305 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1993. Fundamentals.

[44] Liping J. *Quasisecant methods for nonsmooth optimization and their application in sound field reconstruction*. PhD thesis, School of Science, Information Technology and Engineering, University of Ballarat, December 2011.

[45] N. M. S. Karmitsa, M. M. Mäkelä, and M. M. Ali. Limited memory interior point bundle method for large inequality constrained nonsmooth minimization. *Appl. Math. Comput.*, 198(1):382–400, 2008.

[46] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[47] K. C. Kiwiel. *Methods of descent for nondifferentiable optimization*, volume 1133 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1985.

[48] K. C. Kiwiel. A dual method for certain positive semidefinite quadratic programming problems. *SIAM J. Sci. Statist. Comput.*, 10(1):175–186, 1989.

[49] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Programming*, 46(1, (Ser. A)):105–122, 1990.

[50] K. C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim.*, 16(4):1007–1023, 2006.

[51] K. C. Kiwiel. A proximal-projection bundle method for Lagrangian relaxation, including semidefinite programming. *SIAM J. Optim.*, 17(4):1015–1034, 2006.

[52] K. C. Kiwiel. Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM Journal on Optimization*, 18(2):379, 2007.

[53] C. Lemarechal. An extension of Davidon methods to non differentiable problems. *Math. Programming Stud.*, (3):95–109, 1975. Nondifferentiable optimization.

[54] C. Lemaréchal. Nondifferentiable optimization. In *Optimization*, volume 1 of *Handbooks Oper. Res. Management Sci.*, pages 529–572. North-Holland, Amsterdam, 1989.

[55] C. Lemaréchal and J. Zowe. A condensed introduction to bundle methods in nonsmooth optimization. In *Algorithms for continuous optimization (Il Ciocco, 1993)*, volume 434 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 357–382. Kluwer Acad. Publ., Dordrecht, 1994.

[56] R. M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM J. Optim.*, 10(3):917–941 (electronic), 2000.

[57] L. Lukšan and Emilio Spedicato. Variable metric methods for unconstrained optimization and nonlinear least squares. *J. Comput. Appl. Math.*, 124(1-2):61–95, 2000. Numerical analysis 2000, Vol. IV, Optimization and nonlinear equations.

[58] L. Lukšan and J. Vlček. A bundle-Newton method for nonsmooth unconstrained minimization. *Math. Programming*, 83(3, Ser. A):373–391, 1998.

[59] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *J. Optim. Theory Appl.*, 102(3):593–613, 1999.

[60] L. Lukšan and J. Vlček. Introduction to nonsmooth analysis. theory and algorithms. Tech. Rep. DMSIA, Serie Didattica 1/2000, Universita degli Studi di Bergamo, Bergamo, 2000.

[61] L. Lukšan and J. Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. *Technical Report No. 78, Institute of Computer Science, Academy of Sciences of the Czech Republic*, 2000.

[62] L. Lukšan and J. Vlček. Algorithm 811: Nda: Algorithms for nondifferentiable optimization. *ACM Transaction on Mathematical Software*, 27(2):193–213, 2001.

[63] M. M. Mäkelä. Survey of bundle methods for nonsmooth optimization. *Optim. Methods Softw.*, 17(1):1–29, 2002.

[64] M. M. Mäkelä and P. Neittaanmäki. *Nonsmooth optimization*. World Scientific Publishing Co. Inc., River Edge, NJ, 1992. Analysis and algorithms with applications to optimal control.

[65] R. Mifflin. An algorithm for constrained optimization with semismooth functions. *Math. Oper. Res.*, 2(2):191–207, 1977.

[66] R. Mifflin. Semismooth and semiconvex functions in constrained optimization. *SIAM J. Control Optimization*, 15(6):959–972, 1977.

[67] R. Mifflin. A quasi-second-order proximal bundle algorithm. *Math. Programming*, 73(1, Ser. A):51–72, 1996.

[68] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *J. Optim. Theory Appl.*, 142(1):205–228, 2009.

[69] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[70] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1, Ser. A):127–152, 2005.

[71] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Math. Program.*, 120(1, Ser. B):221–259, 2009.

[72] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

[73] E. A. Nurminskiĭ. On the convergence of the method of suitable affine subspaces for solving the problem of the shortest distance to a simplex. *Zh. Vychisl. Mat. Mat. Fiz.*, 45(11):1991–1999, 2005.

[74] E. A. Nurminskiĭ. Projections onto externally specified polyhedra. *Zh. Vychisl. Mat. Mat. Fiz.*, 48(3):387–396, 2008.

[75] E. Polak and J. O. Royset. Algorithms for finite and semi-infinite min-max-min problems using adaptive smoothing techniques. *J. Optim. Theory Appl.*, 119(3):421–457, 2003.

[76] E. Polak, J. O. Royset, and R. S. Womersley. Algorithms with adaptive smoothing for finite minimax problems. *Journal of Optimization Theory and Applications*, 119:459–484, 2003. 10.1023/B:JOTA.0000006685.60019.3e.

[77] E. Polak, J. O. Royset, and R. S. Womersley. Algorithms with adaptive smoothing for finite minimax problems. *J. Optim. Theory Appl.*, 119(3):459–484, 2003.

[78] B. T. Polyak. *Introduction to optimization*. Translations Series in Mathematics and Engineering. Optimization Software Inc. Publications Division, New York, 1987. Translated from the Russian, With a foreword by Dimitri P. Bertsekas.

[79] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.*, 7:155–162, 1964.

[80] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optimization*, 14(5):877–898, 1976.

[81] C. Sagastizábal and M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM J. Optim.*, 16(1):146–169 (electronic), 2005.

[82] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optim.*, 2(1):121–152, 1992.

[83] N. Z. Shor. *Minimization methods for nondifferentiable functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985. Translated from the Russian by K. C. Kiwiel and A. Ruszczyński.

[84] N. Z. Shor. *Nondifferentiable optimization and polynomial problems*, volume 24 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, 1998.

[85] J. Vlček and L. Lukšan. Globally convergent variable metric method for nonconvex nondifferentiable unconstrained minimization. *J. Optim. Theory Appl.*, 111(2):407–430, 2001.

[86] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Math. Programming Stud.*, (3):145–173, 1975. Nondifferentiable optimization.

[87] P. Wolfe. Finding the nearest point in a polytope. *Math. Programming*, 11(2):128–149, 1976.

[88] A. E. Xavier and A. A. F. de Oliveira. Optimal covering of plane domains by circles via hyperbolic smoothing. *J. Global Optim.*, 31(3):493–504, 2005.

[89] S. Xu. Smoothing method for minimax problems. *Comput. Optim. Appl.*, 20(3):267–279, 2001.

[90] Y. F. Yang. A new trust region method for nonsmooth equations. *ANZIAM J.*, 44(4):595–607, 2003.

[91] A. Zaffaroni. Continuous approximations, codifferentiable functions and minimization methods. In *Quasidifferentiability and related topics*, volume 43 of *Nonconvex Optim. Appl.*, pages 361–391. Kluwer Acad. Publ., Dordrecht, 2000.

VITA

PERSONAL INFORMATION

Surname, Name: Tor, Ali Hakan
Date and Place of Birth: 20 June 1982, Ankara
email: hakantor@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
BS	Ankara Universty, Mathematics	2004
High School	Yahya Kemal Beyatlı High School	1998

AWARDS

- **The Scientific and Technical Research Council of Turkey (TUBITAK)**
  - The Integrated Ph.D. Scholarship, 2006-2010

WORK EXPERIENCE

Year	Place	Enrollment
2005-present	METU, Mathematics Department	Research Assistant

FOREIGN LANGUAGES

English (fluent)

PUBLICATIONS

1. A. M. Bagirov, A. N. Ganjehlou, H. Tor, and J. Ugon. A generalized subgradient method with piecewise linear subproblem. *Dyn. Contin. Discrete Impuls. Syst. Ser. B Appl. Algorithms*, 17(5):621–638, 2010.

2. A. M. Bagirov, A. N. Ganjehlou, J. Ugon, and A. H. Tor. Truncated codifferential method for nonsmooth convex optimization. *Pac. J. Optim.*, 6(3):483–496, 2010.

CONFERENCE PAPERS

1. A. H. Tor, B. Karasözen, and A. Bagirov. Truncated codifferential method for linearly constrained nonsmooth optimization. *Vilnius Gediminas Technical University Publishing House "Technika"*, pages 87–93 2010.

INTERNATIONAL CONFERENCE PRESENTATIONS

1. A. H. Tor, A. Bagirov and B. Karasözen. Limited codifferential method for nonsmooth convex optimization problems. *International Conference on Applied and Computational Mathematics (ICACM)*, October 3 - 6, 2012, Ankara, TURKEY.
2. A. H. Tor and A. Bagirov. A Truncated Codifferential Method: Using hypogradients from current and previous iterations. *25th European Conference on Operational Research*, June 8-11, 2012, Vilnius, LITHUANIA.
3. A. H. Tor, B. Karasözen and A. Bagirov. Truncated Codifferential Method for Linearly Constrained Nonsmooth Optimization. *International Conference 24thÂ Mini EURO Conference "Continuous Optimization and Information-BasedÂ Technologies in the Financial Sector" (MEC EurOPT 2010)*, June 23 - 26, 2010, Izmir, TURKEY.
4. A. H. Tor and A. Bagirov. Truncated codifferential method for nonsmooth convex optimization. *23rd European Conference on Operational Research*, July 5 - 8, 2009, Bonn, GERMANY.
5. A. Bagirov, J. Ugon and A. H. Tor. Generalized subgradient algorithm for nonsmooth, nonconvex optimization. *23rd European Conference on Operational Research*, July 5 - 8, 2009, Bonn, GERMANY.

NATIONAL CONFERENCE PRESENTATIONS

1. A. H. TOR, B. Karasözen ve A. Bagirov. Lineer Kısıtlı Türevi Olmayan Konveks Optimizasyon Problemleri için Kesilmiş Codiferansiyel Metot. *XIII. Ulusal Matematik Sempozyumu*, 4-7 Ağustos 2010, Kayseri, Tükiye.

PARTICIPATION OF INTERNATIONAL CONFERENCES

1. 7th EUROPT Workshop "Advances in Continuous Optimization" July 3-4 2009, Remagen, Germany.