SELECTIVE LINEARIZATION FOR MULTI-BLOCK CONVEX OPTIMIZATION

By

YU DU

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Operations Research

written under the direction of

Andrzej Ruszczyński - Xiaodong Lin

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2017

ABSTRACT OF THE DISSERTATION

Selective Linearization for Multi-Block Convex Optimization

By YU DU

Dissertation Director:

Andrzej Ruszczyński - Xiaodong Lin

We consider the problem of minimizing a sum of several convex non-smooth functions. In this thesis, we introduce a new algorithm called the selective linearization method, which iteratively linearizes all but one of the functions and employs simple proximal steps. The algorithm is a form of multiple operator splitting in which the order of processing partial functions is not fixed, but rather determined in the course of calculations. It proposes one of the first operator-splitting type methods which are globally convergent for an arbitrary number of operators without artificial duplication of variables. This algorithm is a multi-block extension of the alternating linearization (ALIN) method for solving structured non-smooth convex optimization problems.

Global convergence is proved and estimates of the convergence rate are derived. Specifically, under a strong convexity condition, the number of iterations needed to achieve solution accuracy ε is of order $\mathcal{O}\big(\ln(1/\varepsilon)/\varepsilon\big)$. The convergence rate analysis technique invented by us can also be used to derive the rate of convergence of the classical bundle method and ALIN method, for which no convergence rate estimate has been available so far.

We report results of extensive comparison experiments in structured regularization problems such as large-scale fused lasso regularization problems and overlapping group lasso problems. The numerical results demonstrate the efficacy and accuracy of the method.

Acknowledgments

I would like to show the deepest appreciation to Professor Andrzej Ruszczyński and Professor Xiaodong Lin, my advisors, for their constant dedication and encouragement in the past seven years. I still remember cheerfully of my time as an student in their classes ever since I was doing Master's degree at Rutgers University, and that was exactly the moment when this journey started. I am truly grateful to them introducing me to the fields of nonsmooth optimization and statistical learning.

Professor Ruszczyński played the most fundamental role in my research and without his patience and openness, this thesis would never have been possible. He gave me extraordinary courage, and guided me to where I am today. He contributed his time, ideas and guided me through the most difficult times in my research. He continuously encouraged me to pursue creative ideas and always stood closely to offer help. He always made himself available to me no matter how busy he was and provided insightful discussions about our research and even shared with me many truths of the life. As my role model, he has taught me invaluable lessons including passion and patience for innovation, commitment to excellence, and skills for presenting ideas and writing reports/papers, which are not only helpful for doing academic research but will also be valuable assets for my career success.

Professor Xiaodong Lin helped me come up with the thesis topic and gave me freedom to pursue various projects, summer internships or personal affairs without objection. He gave me the greatest support for completing my computing tasks and helped me with job searching. I look forward to further collaboration with him in the future.

I also want to thank the members of my Ph.D. committee, Professor Jonathan Eckstein and Professor Darinka Dentcheva for their valuable advice and suggestions about my research.

Additionally, I would like to express my gratitude and appreciation to Professor Endre

Boros and Professor Adi Ben-Isral, Professor Kemal Gursoy and all the faculty members of RUTCOR and some Professors from Business School, Mathematics Department, Statistics Department and Computer Science Department for giving me abundant knowledge, thoughtful guidance and considerate advice during my graduate studies. I want to show my great dedication to the memory of Professor András Prékopa. Not only was he a true scholar, but also our beloved mentor.

Dozens of people have helped and taught me immensely at Rutgers, I would like to express my appreciation to them. My time at RUTCOR and later MSIS Department was made enjoyable in large part due to many friends. I am grateful for time spent with Wang Yao, Anh Ninh, Mohammad Ranjbar, Minh Pham, Gyorgy Matyasfalvi, Marta Cavaleiro, Javier Rubio Herrero, Peter Mursic, Jingnan Fan, Jianing Yao, Kaicheng Wu, Jinwook Lee, Joonhee Lee, Emre Yamangil, and Tsvetan Asamov. I will forever remember our times of study and discussion.

I also thank my friends for providing support and friendship that I needed. I especially thank Hanlong Fang, Sijian Tang, and Zhan Li from Math Department, Menglin Jiang, Meng Li from Computer Science Department and Ting Yang, Yifan Zhang, and Yunqing Hu from Statistics Department for being supportive throughout my time at Rutgers and for discussing various interesting subjects that are related to our own research.

I also thank RUTCOR and MSIS staff Clare Smietana, Terry Hart, Lynn Agre, and Arleen Verendia for their kind support. They have all been so friendly and personable, making me feel like a family member of RUTCOR.

I would like to acknowledge the financial support that I received by the Excellence Fellowship and Graduate Assistantship awards at Rutgers University, that made my Ph.D. work.

Most of all, I would like to thank my family. My mother Linping Gao and father Zhongsheng Du raised me up with lots of love and hard work. I owe them more than words can express. They provided greatest support for my study abroad. They sacrificed so much of their life for me. They taught me the most important things in my life and has been an outstanding inspiration to me. Without their motivation and strength I would have never had the courage to overcome the adversities I have faced. They are the main reason for

many things I have done, to make them proud. There are no words to convey how much I love them. My family have cherished with me every great moment, they are the most basic sources of my energy.

Much appreciated!

Du, Yu

Edison, New Jersey

March, 2017

Dedication

To my parents, for their constant support and unconditional love!

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Preliminaries

1.1 Introduction

The topic of this thesis is a large-scale optimization method for minimizing a sum of many convex non-differentiable functions. In the big data era, we have seen extensive development of the theory and methods for *structured regularization*, one of the most fundamental techniques to address the "big data" challenge. The basic problem is to minimize the following objective function with two components (blocks):

$$\mathcal{F}(x) = f_1(x) + f_2(x)$$

where $f_1(\cdot)$ is the loss function and $f_2(\cdot)$ is a penalty function that imposes *structured regularization* to the model.

Many data mining and machine learning problems can be cast within this framework, and many efficient methods can solve these problems including *operator splitting* methods see [DR56, BC11, Com09, EB92, LM79], and their dual versions, known as *Alternating Direction Methods of Multipliers* (ADMM) (see, [GM76, GM75, GT89]). The Alternating Linearization method(ALIN) [KRR99] handles two-block convex problems by introducing an additional improvement test to the operator splitting methods and it adapts some ideas of bundle methods of nonsmooth optimization [HUL93, Kiw85, Rus06]. The recent application of ALIN to structured regularization problems in [LPR14] is proved to be very successful, with fast convergence, good accuracy, and scalability to very large dimensions. It may be worth noticing that the recent application of the idea of alternating linearization by [GMS13] removing the update test from the method of [KRR99], thus effectively reducing it to an operator splitting method.

Most existing techniques for structured regularization are developed under the two-block

framework. It is shown that direct generalization of ADMM to three or more blocks may fail to converge [CHYY14]. A known way is to introduce N copies $x^1 = x^2 = \cdots = x^N$ of x, and reduce the problem to the two-function case in the space $\mathbb{R}^{nN}$ [CP11]:

$$\min \sum_{i=1}^{N} f_i(x^i) + I(x^1, \ldots, x^N)$$

with $I(\cdot)$ denoting the indicator function of the subspace $x^1 = x^2 = \cdots = x^N$. Similar ideas were used in stochastic programming, under the name of *Progressive Hedging* [RW91].

We extend the ALIN framework to optimization problems involving multiple components. Namely, we aim to minimize

$$\mathcal{F}(x) = f_1(x) + \sum_{i=2}^{N} f_i(x),$$

where the penalty function is a sum of multiple components. This type of generalization has many practical applications, in introducing sparsity, block-sparsity, network structure, dynamic structure, low-dimensional Fourier representation, etc. to the learning tasks.

We introduced a new algorithm called the *Selective Linearization Method* (SLIN). It generates a sequence of points $x^k \in \mathbb{R}^n$ with a monotonic sequence of corresponding function values $\mathcal{F}(x^k)$. At each iteration, it linearizes all but one of the component functions and uses a proximal term penalizing the distance to the last iterate. The order of processing the functions is not fixed; the method uses a precise criteria for selecting the function to be treated exactly at the current step. It also employs special rules for updating the proximal center. These two rules differ our approach from the simultaneously proposed incremental proximal method of [Ber15], which applies to smooth functions only, and achieves linear convergence rate in this case.

This thesis contains several original contributions to the theory and practice of large-scale non-smooth optimization:

1. It proposes one of the first operator-splitting type methods *SLIN* which are globally convergent for an arbitrary number of operators (subdifferentials of the said functions), without artificial duplication of variables. This surprising result has been obtained thanks to the idea of determining the order of splitting on-line, depending on the

values of the functions minimized, and accepting the result of the splitting step only when it leads to the decrease of the overall objective.

2. It contains not only the proof of global convergence but also the proof of convergence rate, which is a new contribution even in the case of two blocks. In fact, the technique invented by us can be also used to derive the rate of convergence of the classical bundle method, for which no rate estimate has been available so far.

3. The thesis provides extensive experimental results for very large problems, which demonstrate the efficacy and accuracy of the method. We have done extensive comparison experiments in fused lasso regularization problems and overlapping group lasso with tree structure, fixed order and random order cases. The experimental results show that the method proposed in this thesis is the best general-purpose method for multi-block non-smooth optimization in practice.

1.1.1 Outline of the dissertation

The rest of this chapter will give an introduction on the problem of interest, problem formulation and our contribution. Chapter 2 provides the literature review on several existing operator splitting methods, especially ALIN method which lays the foundation to our work. Chapter 3 introduces the idea of bundle method for versions with multiple cuts and with cut aggregation. Rate of convergence is derived, which is our new contribution. Chapter 4 introduces the idea of selective linearization method for solving multi-block non-smooth optimization problems. Global convergence is proved and estimates of the convergence rate are derived. In Chapter 5, we illustrate SLIN's operation on structured regularized regression problems involving many blocks. Comprehensive experiments show that SLIN is a highly efficient and reliable general-purpose method for multi-block optimization of convex non-smooth functions. Conclusion and future research plans are finally discussed in Chapter 6.

1.2 Motivating examples and problem formulation

1.2.1 Motivating examples of multi-block structured regularization problems

In many areas in data mining and machine learning, such as computer vision and compressed sensing, bio-informatics and remote sensing, we encounter the big data challenge. The big data concept can be understood in two ways. First, the data size is quite large. For example, we may have millions of data records to process. The data can also come in a streaming way, such as online YouTube videos. Second, the data can be of high dimension and the sample size is much smaller than the dimension. For instance, in the RNA sequence Figure 1.1, the sample size is about $10,000$ times smaller than the dimension [DHHH13]. By using classical regression models, such as the square loss function, it is not possible to estimate the desired features with limited number of samples.

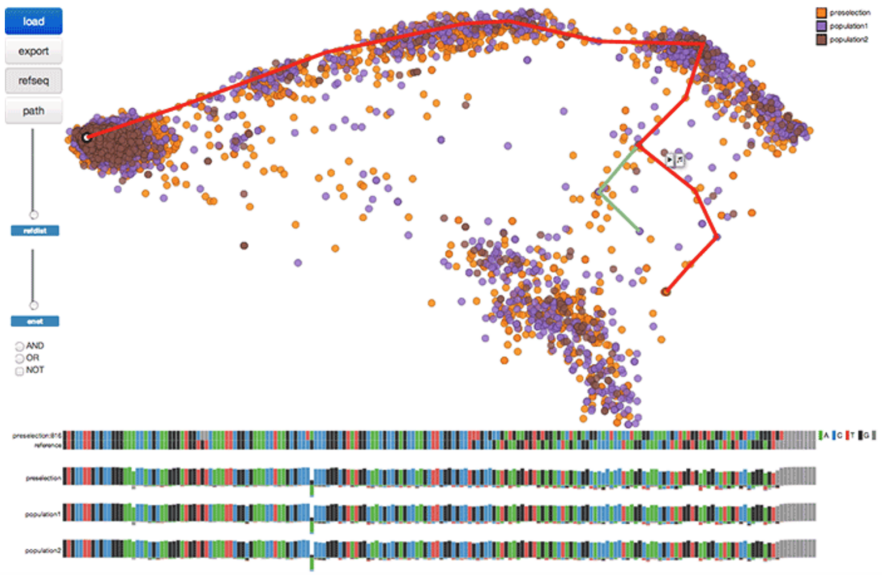

Figure 1.1: The RNA Sequence [DHHH13]

In recent years, we have seen extensive development for modeling the high dimensional statistic problems by incorporating complex structured regularization penalties into the model. The resulting optimization model (1.1) consists of a sum of convex loss functions f_i, which measures the goodness-of-fit of the data, and multiple convex regularization functions

(or penalties) h_j.

$$\min_{x\in\mathbb{R}^n} F(x) = \sum_{i=1}^{M} f_i(x) + \sum_{j=1}^{N} h_j(B_j x) \tag{1.1}$$

This model is proven useful to solve these high dimensional statistical problems. From the model, all the functions are convex but not necessarily smooth.

Compressed MRI

The first example of structured regularization problem is a medical compressed MR image recovery problem, where we want to recover the true MR image from a noised scanning image. Magnetic Resonance (MR) imaging has been widely used in medical diagnosis because of its non-invasive manner and excellent depiction of soft tissue changes. Recent developments in compressed sensing theory show that it is possible to accurately reconstruct the magnetic resonance images from for example only 20% sampling data and therefore significantly reduce the scanning duration. This is due to the fact that in MRI the measurement is very expensive with few sensors; the sensing process is very slow as well see Figure 1.2. By quickly sampling 20% of the data and accurate reconstructing the image, we can save lives of many people in emergency [ZGWY15].

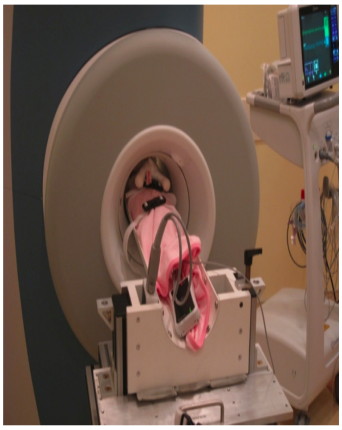

Figure 1.2: MRI sensing [ZGWY15]

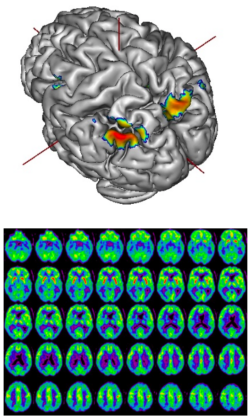

Figure 1.3: MRI [ZGWY15]

For recovering and deblurring the magnetic resonance images, a model is suggested to obtain a high quality restored image using total variation penalty and wavelet based penalty. The objective function is a linear combination of three blocks: a loss function, a total variation (TV) norm and L_1 norm with wavelet transformation on the restored image

signal (1.2):

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}||Rx-b||^2 + \alpha||x||_{TV} + \beta||\Phi x||_1. \tag{1.2}$$

In the above formulation, R is a partial Fourier transform, Φ is the wavelet transform, b is the vector of under-sampled Fourier measurements, α and β are two positive parameters. The problem seeks to reconstruct the image signal x given the sampling measurement b and the sampling matrix R. The TV norm has the effect of deblurring, which means reducing significant differences between neighboring pixels. The L_1 norm has the filtering effect, extracting significant coefficients from the wavelet transformation. Combined with the wavelet based penalty, the solution of the problem is not only a high quality restored image but is also sparse in the wavelet domain. This model has been shown to be one of the most powerful models for the compressed MR image recovery. However, with high dimensional images, the formulation is very challenging. More importantly, the TV and L_1 with wavelet norms are complex non-separable and non-smooth structured penalties, which makes this problem difficult to solve.

Sparse and low rank matrix reconstruction

The second example is the Netflix problem in the area of recommendation systems. In Netflix website, some users submit ratings on different movies based on the users' preferences. But for each movie, there are only limited number of ratings. Some users only rate a few movies based on their own preference. What we observe in the rating matrix is a very sparse matrix with many missing values, as in Figure 1.4. Netflix company would like to predict the remaining entries in order to make good recommendations to customers on what to watch next.

The task is to complete this sparse matrix so that the Netflix will use completed matrix to recommend new movies to existing users. In this case, we build up a model to solve this matrix completion problem. The model is the following three block matrix regularization problem (1.3).

$$\min_S ||P_\Omega(S) - P_\Omega(A)||_F^2 + \gamma||RS||_1 + \tau||S||_*, \tag{1.3}$$

where S is the reconstructed matrix, and A is observed matrix. $P_\Omega(A)$ is the projection onto

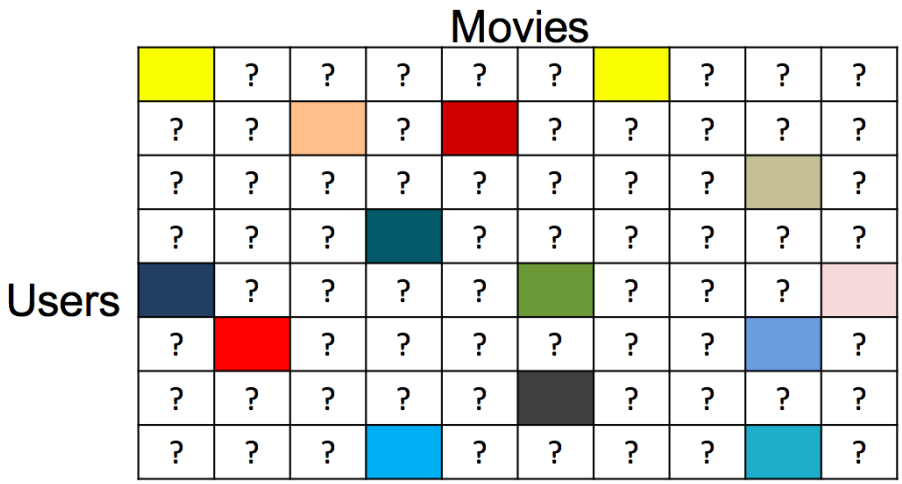

Figure 1.4: The Netflix Problem [ZGWY15]

the observed entries such that missing values are set to zero. The first square loss term, forces the difference of projection between the completed matrix and original observed matrix not far away from each other. The second term is the L_1 norm of the matrix S times R. This R matrix has the property that it groups the users with similar types of preferences together. For example, they may like the same type of movies or dislike the same type of movies. The third term is the nuclear norm of the matrix S. We assume that the ratings matrix is low-rank, since users' preference can often be described by a few factors, such as the movie genre and time of release, etc. In statistical learning, one may apply the regularization penalty in the form of a nuclear norm promoting low-rank solutions. Again L_1-norm and nuclear norm are complex non-separable and non-smooth structured matrix penalties, which makes this problem difficult to solve.

Fused lasso model in CGH analysis

The third example is the CGH data analysis in bio-informatics. In Figure 1.5, the horizontal line represents the DNA sequence. Each grey point represents the CGH signal at one gene location. The CGH signals are useful techniques to measure the differences between numbers of gene copies for solid tumor cells and numbers of gene copies for normal cells. If CGH signals are zero, it corresponds to the normal copy of gene cell. If the CGH signals are nonzero, it is more likely to be an irregular gene copy of tumor cell.

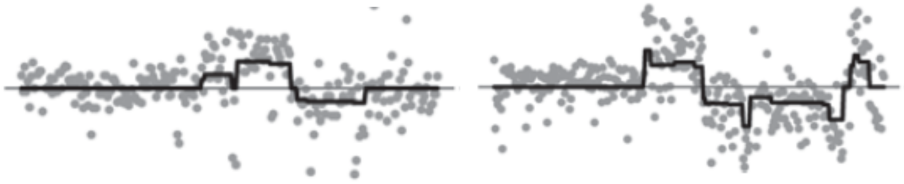

Figure 1.5: Array CGH Data Analysis [TW07]

The problem is to denoise the CGH signals to a piecewise black line which has relatively sparse areas with nonzero values. The resulting structured regularization model is the following fused lasso model of [TW07] which deals with the CGH detection problem (1.4):

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}||Ax-b||^2 + \lambda_1 \sum_{j=1}^{p} ||x||_1 + \lambda_2 \sum_{j=1}^{p-1} ||x_{j+1}-x_j||_1 \tag{1.4}$$

The first term is the square loss, making the estimated signals not far away from the observed signals. The model is useful for determining which areas of the signal are likely to be nonzero by adding the L_1 sparsity norm (forcing most signals to be zeros). The fused lasso shrinks the differences of the signals in consecutive locations to zero.

This problem involves a multi-block non-separable and non-smooth L_1 and fused lasso penalties in large scale, which makes it difficult to solve.

Overlapping group lasso in text mining

Another lasso-type example is the overlapping group lasso model, with a composite of overlapping group lasso regularizers:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2K\lambda}\left\|b-Ax\right\|_2^2 + \sum_{j=1}^{K} d_j \left\|x_{\mathcal{G}_j}\right\|_2, \tag{1.5}$$

where $A \in \mathbb{R}^{m\times n}$. This group lasso regularizer has been proven useful in high-dimensional problems with the capability of selecting meaningful groups of features.

This model contains the first function as $f(x) := \frac{1}{2K\lambda}||b-Ax||_2^2$ where parameter $\lambda > 0$ and the number of groups K are pre-specified parameters. The second part is a sum of regularization terms, where each penalty function $h_j(x) = d_j||x_{\mathcal{G}_j}||_2$ and the weights $d_j > 0$ are known parameters. $\mathcal{G}_j \subseteq \{1,\ldots,p\}$ is the index set of a group of variables and $x_{\mathcal{G}_j}$ denotes the subvector of x with coordinates in $\mathcal{G}_j$.

The features in groups can overlap as needed. The groups can overlap in a tree structured order. For example in Figure 1.6, one node corresponds to one group, one group can be a subset of another group, or disjoint with the other groups. Since this overlapping regularization term is not separable, this problem is still a multi-block convex optimization problem.

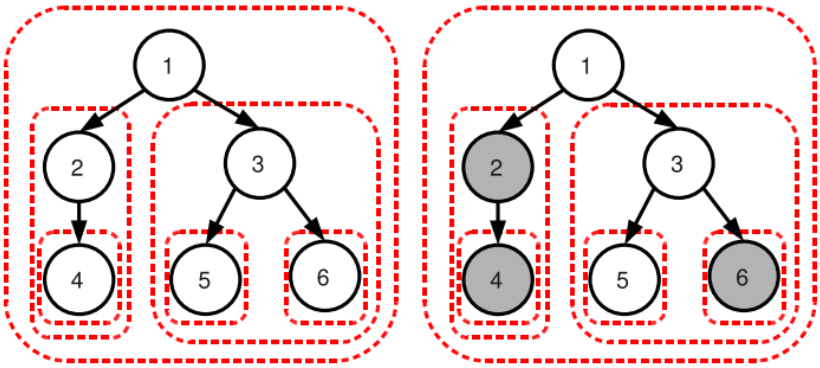

Figure 1.6: Tree structured overlapping group lasso [JMOB11a]

Tree structured order overlapping group lasso is widely used in hierarchic dictionary learning. It's a powerful technique in uncovering the tree structured sparsity over the features. Hierarchies selections, typically used in neural networks and deep learning architectures [Ben09] have emerged as a natural structure in several applications in topic modeling of text documents. Topic modeling is a method to discover abstract "topics" that occur in a collection of documents. It is a frequently used text mining tool for the discovery of hidden semantic structures in a text body. In Figure 1.7, a sentence always has a syntax structure. Based on the the syntax structure, one could impose the overlapping group lasso structure to the learning process. Consequently, we can identify the important topics from a collection of words in the document.

1.2.2 Problem formulation for multi-block convex optimization

For these mentioned applications, we notice that their objective functions all share the same form:

$$\min_{x \in \mathbb{R}^n} f_1(x) + f_2(x) + \ldots + f_N(x), \tag{1.6}$$

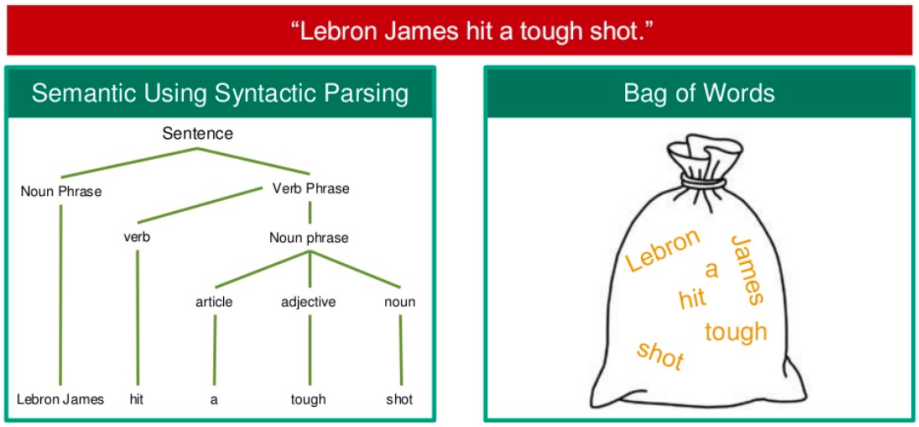

Figure 1.7: Text mining: topic modeling [Kwa15]

where f_1,f_2,...,f_N: $\mathbb{R}^n \to \mathbb{R}$ are convex but not necessarily smooth functions. This is the the general problem formulation of our optimization problems. In this thesis, we introduce a new algorithm called SLIN to solve this multi-block convex optimization. More importantly, we obtain competitive convergence results. Specifically, we guarantee global convergence and almost $O(1/k)$ convergence rate with only 1 out of N functions being strongly convex, where k is the iteration number.

Chapter 2

Review of Related Existing Methods

2.1 Operator splitting methods

Selective linearization algorithm comes from the idea of proximal point algorithm [Roc76] and operator splitting methods for two-block convex programming half a century ago. Its roots are primarily in functional analysis, engineering problems and mathematical physics. As the rising attention of signal processing and machine learning, operator splitting methods have many more applications.

Suppose that we are trying to solve the following problem:

$$\min_{x\in\mathbb{R}^n} f(x) + h(x). \tag{2.1}$$

The solution for this problem is $\hat{x}$ such that:

$$0 \in \partial f(\hat{x}) + \partial h(\hat{x}).$$

We can consider two subdifferentials as two maximum monotone operator M_1 and M_2 such that:

$$0 \in (M_1 + M_2)(\hat{x}) \tag{2.2}$$

given the two monotone operators on the space $\mathbb{R}^n$.

The operator splitting methods originated from Peaceman-Rachford method (2.3) [PR55] and Douglas-Rachford method (2.4) [DR56]. [LM79] analyzed the methods for finding a zero of the sum of two maximum monotone operator (2.2), and later developed and analyzed by [EB92, Com09, BC11], among others.

$$x_{k+1} = (I + \rho M_2)^{-1}(I - \rho M_1)(I + \rho M_1)^{-1}(I - \rho M_2)x_k. \tag{2.3}$$

$$x_{k+1} = (I + \rho M_2)^{-1}[(I + \rho M_1)^{-1}(I - \rho M_2) + \rho M_2]x_k. \tag{2.4}$$

In formulas (2.3) and (2.4), x_k is the generated sequence starting from any point $x_1 \in \mathbb{R}^n$. I is an identity matrix. M_1 and M_2 are monotone operators. ρ is a positive number.

For solving (2.1), Peaceman-Rachford method works as follows [LM79].

Peaceman-Rachford

1: repeat
2: $\tilde{x}_h \leftarrow argmin\{\tilde{f}(x) + h(x) + \frac{1}{2\rho}||x - \hat{x}||^2\}$.
3: $g_h \leftarrow -g_f - \frac{1}{\rho}(\tilde{x}_h - \hat{x})$
4: $\hat{x} \leftarrow \tilde{x}_h$
5: $\tilde{x}_f \leftarrow argmin\{f(x) + \tilde{h}(x) + \frac{1}{2\rho}||x - \hat{x}||^2\}$.
6: $g_f \leftarrow -g_h - \frac{1}{\rho}(\tilde{x}_f - \hat{x})$
7: $\hat{x} \leftarrow \tilde{x}_f$
8: until (Stopping Test)

Peaceman-Rachford method involves *proximal steps* in step 2 and step 5 and updates proximal center $\hat{x}$ after every proximal step. However, it can not guarantee convergence for general two maximum monotone operators.

According to [LM79], Douglas-Rachford splitting scheme works as follows.

Douglas-Rachford

1: repeat
2: $\tilde{x}_h \leftarrow argmin\{\tilde{f}(x) + h(x) + \frac{1}{2\rho}||x - \hat{x}||^2\}$.
3: $g_h \leftarrow -g_f - \frac{1}{\rho}(\tilde{x}_h - \hat{x})$
4: $\tilde{x}_f \leftarrow argmin\{f(x) + \tilde{h}(x) + \frac{1}{2\rho}||x - \hat{x}||^2\}$.
5: $g_f \leftarrow -g_h - \frac{1}{\rho}(\tilde{x}_f - \hat{x})$
6: $\hat{x} \leftarrow \tilde{x}_f$
7: until (Stopping Test)

It involves proximal steps in step 2 and step 5 and only updates proximal center $\hat{x}$ at step 6. As the roles of f and h can be switched, the method in which updates are carried always after step 3, but never after step 5 is also equivalent to a scaled DouglasRachford method.

There are also versions of Douglas-Rachford splitting with under and over relaxation parameter $\lambda \in [0, 2]$.

Douglas-Rachford with relaxation

1: repeat
2: $\tilde{x}_h \leftarrow argmin\{\tilde{f}(x) + h(x) + \frac{1}{2\rho}||x - \hat{x}||^2\}$.
3: $g_h \leftarrow -g_f - \frac{1}{\rho}(\tilde{x}_h - \hat{x})$
4: $\tilde{x}_f \leftarrow argmin\{f(x) + \tilde{h}(x) + \frac{1}{2\rho}||x - \hat{x}||^2\}$.
5: $g_f \leftarrow -g_h - \frac{1}{\rho}(\tilde{x}_f - \hat{x})$
6: $\hat{x} \leftarrow (1 - \lambda)\hat{x} + \lambda\tilde{x}_f$
7: until (Stopping Test)

The only difference is to incorporate relaxation parameter when updating the proximal center. When $\lambda \in [0, 1]$, it is under relaxation. When $\lambda \in [1, 2]$, it is over relaxation. It can guarantee global convergence for two block convex problems. In practice, Douglas-Rachford splitting with under and over relaxation parameters can be better or worse than the regular Douglas-Rachford splitting method.

Operator splitting methods are not monotonic with respect to the values of the objective function. Their convergence is based on monotonicity with respect to the distance to the optimal solution of the problem [LM79, EB92].

In chapter 5, we shall compare SLIN algorithm with Douglas–Rachford operator splitting method of [LM79] carrying different relaxation parameters λ and proximal parameters ρ for multi-block convex examples. However, there is no convergence result for multi-block non-separable Douglas–Rachford operator splitting method.

2.2 Alternating linearization method

The Alternating Linearization Method (ALIN) [KRR99] introduced an improvement test to the operator splitting methods, adapted some ideas of bundle methods of nonsmooth optimization [HUL93, Kiw85]. The way of proving its global convergence, is due to [Rus86]. ALIN [LPR14] has been successfully applied to solve two-block structured regularization problems.

However, the convergence rate has been an open question for many years due to the difficulty of analyzing the improvement test. In this thesis, we managed to extend the ALIN algorithm to multi-block SLIN algorithm and solved the open question of convergence rate

for both the bundle methods and the SLIN method.

Since SLIN algorithm is a multi block extension of ALIN algorithm which aims at solving two block structure regularization problems, we present an overview of the ALIN algorithm as follows for solving problem (2.1).

Algorithm Alternating Linearization (ALIN)

1: repeat
2: $\tilde{x}_h \leftarrow argmin\{\tilde{f}(x) + h(x) + \frac{1}{2}||x - \hat{x}||^2_D\}$.
3: $g_h \leftarrow -g_f - D(\tilde{x}_h - \hat{x})$
4: if (Update Test for $\tilde{x}_h$) then $\hat{x} \leftarrow \tilde{x}_h$ end if
5: $\tilde{x}_f \leftarrow argmin\{f(x) + \tilde{h}(x) + \frac{1}{2}||x - \hat{x}||^2_D\}$.
6: $g_f \leftarrow -g_h - D(\tilde{x}_f - \hat{x})$
7: if (Update Test for $\tilde{x}_f$) then $\hat{x} \leftarrow \tilde{x}_f$ end if
8: until (Stopping Test)

Instead of solving problem (2.1), the first step involves minimizing an approximation model of the original problem. It linearizes the component function $f(x)$, keeps $h(x)$, and uses a proximal term penalizing for the distance to the last iterate. The norm $\|x\|_D = \big(\langle x, Dx\rangle\big)^{1/2}$ with a positive definite matrix D, which leads to major computational simplifications. From optimal condition of the above minimization problem, we obtain the subgradient of the component function $h(x)$. At the next iteration, it uses the subgradient information to get another approximation model of the original problem. It keeps the function $f(x)$, linearizes the component function $h(x)$, and adds a proximal term. The algorithm uses a special update step to decide if it is good enough to switch the proximal center. It continues the process until passing a stopping test to achieve an approximate optimal solution.

ALIN algorithm has a special update rule, which guarantees the global convergence and monotonically decreasing of the objective function values.

In [LPR14], ALIN is compared with algorithms including alternating direction method of multipliers(ADMM). It shows that the convergence of ALIN is monotonic, and ALIN has better computational time compared with ADMM, which does not have descent properties, and whose tail convergence may be slow.

2.3 Alternating direction method of multipliers and its multi-block extensions

The dual versions of operator splitting methods are known as Alternating Direction Methods of Multipliers (ADMM) for minimizing the sum of two convex functions ([GM76, GM75, GT89]). It was first observed in [GM76] that the ADMM algorithm can be derived from an application of the Douglas-Rachford algorithm to the dual of (2.1).

To introduce ADMM method, first we can rewrite (2.1), introducing an additional decision variable vector $y \in \mathbb{R}^m$ as follows:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) + h(y) \\ &\text{s.t.} \quad x - y = 0. \end{aligned} \tag{2.5}$$

In the two-block structured regularization problems when $h(y) = g(Mx)$ with some fixed matrix M, the convenient problem formulation is

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) + h(y) \\ &\text{s.t.} \quad Mx - y = 0. \end{aligned} \tag{2.6}$$

ADMM for solving (2.6) takes the following form, for some scalar parameter $c > 0$:

$$\begin{aligned} &x_{k+1} \in \text{argmin}_{x\in\mathbb{R}^n}\{f(x) + g(y_k) + \langle\lambda_k, Mx - y_k\rangle + \frac{c}{2}||Mx - y_k||^2\}, \\ &y_{k+1} \in \text{argmin}_{y\in\mathbb{R}^m}\{f(x_{k+1}) + g(y) + \langle\lambda_k, Mx_{k+1} - y\rangle + \frac{c}{2}||Mx_{k+1} - y||^2\}, \\ &\lambda_{k+1} = \lambda_k + c(Mx_{k+1} - y_{k+1}). \end{aligned}$$

ADMM has been shown to have $O(\frac{1}{k})$ rate of convergence for two blocks convex problems, where k stands for the number of iteration [HY15]. A recent tutorial paper [BPC$^+$10] raised interest in ADMM in the field of large scale distributed optimization. ADMM has been extensively studied in recent years due to its ease of applicability and empirical performance (see, *e.g.*, [BPC$^+$10, CP11], and the references therein). Sometimes, these methods are called *split Bregman methods* (see, *e.g.*, [GO09, YX11]).

More importantly, many people are interested in extending two block ADMM to multi-block ADMM for solving multi-block convex optimization problems, which is the focus here.

However, according to [CHYY14], the standard ADMM for N-block ($N > 2$) convex optimization problems is not necessarily convergent. In response to this negative result, [HL17] and [LMZ15], among others, imposed additional assumptions on the objective functions in order to attain convergence for multi-block ADMM. For example, functions need to be smooth, or at least $N-2$ functions are strongly convex. Others added some restrictions such as limiting the step size in updating the Lagrangian multiplier for the multi-block ADMM method.

However, problem (1.6), the focus of the dissertation, has a simple multi-block structure that enables the use of two-block ADMM with good convergence results. This is discussed in [EY15] and [CP11] for a slightly more generalized version of problem (1.6) and here we briefly review the technique in [EY15] to solve problem (1.6). By variable duplication, we can re-formulate problem (1.6) into the following form:

$$\begin{aligned} \min_{x_1,\ldots,x_N \in \mathbb{R}^n} & \sum_{i=1}^{N} f_i(x_i) \\ s.t.\ & x_1 = x_N, \\ & x_2 = x_N, \\ & \ldots, \\ & x_{N-1} = x_N. \end{aligned} \tag{2.7}$$

The constraints in the reformulated problem (2.7) can be unified into a single constraint in the equation form:

$$\sum_{i=1}^{N} A_i x_i = b,$$

where each $A_i \in \mathbb{R}^{(N-1)n \times n}$ and $b = 0 \in \mathbb{R}^{(N-1)n}$. Each A_i can be viewed as a vertical concatenation of $N-1$ n-by-n matrices. And for $i = 1, \ldots, N-1$, all blocks of the n-by-n matrices are zero matrices except the ith block being an n-by-n identity matrix. For A_N, it consists of $N-1$ n-by-n negative identity matrix. In [EY15], they add redundancy to the above single-equality constraint by introducing an additional set of decision variables

$z_1, \ldots, z_N \in \mathbb{R}^{(N-1)n}$ such that:

$$A_i x_i = z_i \ i = 1, \ldots, N$$
$$\sum_{i=1}^{N} z_i = 0.$$

Then they construct the above formulation into a two-block pattern by concatenating the $x_1, \ldots, x_N$ into a single x variable, and concatenating $z_1, \ldots, z_N$ into a single z variable, and apply two-block ADMM directly. They show that both sub-problems for the x variable and the z variable in two-block ADMM are decomposable and can be solved efficiently. The final complete procedure of applying two-block ADMM to solve problem (2.7) is summarized as follows [EY15]:

$$x_i^{k+1} \in \text{argmin}_{x_i \in \mathbb{R}^n} \{f_i(x_i) + \langle \lambda^k, A_i x_i - z_i^k \rangle + \frac{c}{2} ||A_i x_i - z_i^k||^2\} \ i = 1, \ldots, N$$
$$r^{k+1} = \left(\sum_{i=1}^{N} A_i x_i^k \right)$$
$$z_i^{k+1} = A_i x_i^{k+1} - (\frac{1}{p}) r^k$$
$$\lambda^{k+1} = \lambda^k - (\frac{c}{p}) r^{k+1}.$$

In the above procedure, c and p are two positive parameters.

There are other extensions of ADMM method for solving multi-block convex optimization problems. [CDZ15] introduced an augmented lagrangian method for distributed optimization problems. Other methods of PDMM by [WBL14], sADMM or Jacobian ADMM by [DLPY13] for multi-block convex optimization do not guarantee convergence, although they appears to work well in many practical examples. In Chapter 5 of numerical illustration, we will compare SLIN with those methods.

[CE16] introduced the new block-iterative operator splitting algorithms. [Eck17] introduced an asynchronous algorithm resembling the multi-block ADMM for multi-block convex optimization and parallel computing, which guarantees global convergence. The overall approaches were based on earlier work of projective splitting methods for sums of maximal monotone operators in [ES09].

2.4 Other multi-block nonsmooth optimization methods

A different operator splitting method is the primal-dual splitting scheme for sums of composite parallel-sum type operators [CP11]. Similar methods are [Vu13] and [Con13] with established convergence results.

According to [Vu13], let $\mathbb{R}^n$ and $\mathbb{R}^m$ be finite-dimensional spaces. Let $f \in \Gamma_0(\mathbb{R}^n)$ and $g_i \in \Gamma_0(\mathbb{R}^m)$. Let m be a strictly positive integer, let $(w_i)_{1\leq i\leq m}$ be real numbers in $[0,1]$ such that $\sum_{i=1}^m w_i = 1$. For some $v_i \in [0,\infty]$, suppose that $L_i : \mathbb{R}^n \to \mathbb{R}^m$ is a nonzero matrix.

Consider the primal problem:

$$\min_{x\in\mathbb{R}^n} f(x) + \sum_{i=1}^m w_i g_i(L_i x). \tag{2.8}$$

Many multi-block structured regularization problems can be cast into this framework.

The dual problem is

$$\min_{v_1,\ldots,v_m\in\mathbb{R}^m} f^*(-\sum_{i=1}^m w_i L_i^T v_i) + \sum_{i=1}^m g_i^*(v_i). \tag{2.9}$$

Note that $g_i^*(\cdot) = \max_x\{\langle\cdot, x\rangle - g(x)\}$ is the conjugate function.

The $(x_n)_{n\in N}$ and $(v_{1,n}, ...v_{m,n})_{n\in N}$ are sequences generated by the following routine. It's proved in [Vu13] that (x_n) converges to the primal optimal solution $\hat{x}$ and $(v_{1,n}, ...v_{m,n})$ converges to the dual optimal solution $(\hat{v}_{1,n}, ...\hat{v}_{m,n})$.

Vu's splitting

---

for $n = 1, \ldots, N$
$p_n = \text{prox}_{\tau f}(x_n - \tau(\sum_{i=1}^m w_i L_i^T v_{i,n}))$
$y_n = 2p_n - x_n$
$x_{n+1} = x_n + \lambda_n(p_n - x_n)$
    for $i = 1, .., m$
    $q_{i,n} = \text{prox}_{\sigma_i g_i^*}(v_{i,n} + \sigma_i(L_i y_n))$
    $v_{i,n+1} = v_{i,n} + \lambda_n(q_{i,n} - v_{i,n})$.
    end for
end for

Note that

$$\text{prox}_{g^*}(v) = \arg\min_u [g^*(u) + \frac{1}{2}||u - v||^2]. \tag{2.10}$$

The minimization problem (2.10) can be manipulated as:

$$\begin{aligned}
&\min_u[g^*(u) + \frac{1}{2}||u-v||^2] \\
&= \min_u\{\max_x\{\langle u, x\rangle - g(x)\} + \frac{1}{2}||u-v||^2\} \\
&= \max_x\{\langle v-x, x\rangle - g(x) + \frac{1}{2}||x||^2\}, (u = v-x) \\
&= \max_x\{\langle v, x\rangle - g(x) - \frac{1}{2}||x||^2\} \\
&= \min_x[g(x) + \frac{1}{2}||x-v||^2].
\end{aligned} \tag{2.11}$$

One can see that the minimizer to the problem (2.11) in the last equation is $prox_g(v)$, which is similar to the proximal steps in ALIN method.

We shall compare Vu's splitting method with other methods our numerical illustration chapter in solving structured regularization problems.

Chapter 3

The Convergence Rate of Bundle Methods

We prove that the bundle method for nonsmooth optimization achieves solution accuracy ε in at most $\mathcal{O}\big(\ln(1/\varepsilon)/\varepsilon\big)$ iterations, if the function is strongly convex. The result is true for the versions of the method with multiple cuts and with cut aggregation.

3.1 Introduction

The objective of this chapter is to provide a worst-case bound on the rate of convergence of the bundle method for solving convex optimization problems of the following form:

$$\min_{x\in\mathbb{R}^n} F(x), \tag{3.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}$ is a convex function. The only additional assumption about the function needed to bound the rate is strong convexity of the function about the minimum point.

The bundle methods were developed in [Lem78, Mif82]. First rigorous convergence analysis and versions with cut aggregation were provided in [Kiw83, Kiw85, Rus86]. For a comprehensive treatment of bundle and trust region methods, see [BGLS03, HUL93]. Although the bundle method is a method of choice for nonsmooth optimization, no general rate of convergence results are available. This is due to the complicated structure of the method, in which successive iterations carry out different operations, depending on the outcome of a sufficient descent test.

Some results on the rate of convergence are available for the related bundle level method [LNN95], which achieves $O(1/\epsilon^2)$ iteration complexity for general nonsmooth convex programming problems. Similar results have been obtained for modified versions in [Kiw95] and [Lan15].

Our contribution is to prove at most $\mathcal{O}\big(\ln(1/\varepsilon)/\varepsilon\big)$ iteration complexity of the classical

bundle method, under the condition of strong convexity about the minimum point. This is achieved by bounding the numbers of null steps between successive descent steps, and integrating these bounds across the entire run of the method. The result holds true for two versions of the method: with multiple cuts and with cut aggregation.

In section 3.2, we present both versions of the bundle method and recall its convergence properties. Section 3.3 contains several auxiliary results. A worst-case bound on the convergence rate of the method is derived in section 3.4.

We use $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ to denote the usual scalar product and the Euclidean norm in a finite dimensional space.

3.2 The Bundle Method

The bundle method is related to the fundamental idea of the *proximal point method*, which uses the *Moreau–Yosida regularization* of $F(\cdot)$,

$$F_\rho(y) = \min_x \Big\{ F(x) + \frac{\rho}{2} \|x - y\|^2 \Big\}, \quad \rho > 0, \tag{3.2}$$

to construct the *proximal step* for (3.1),

$$\operatorname{prox}_F(y) = \arg\min_x \Big\{ F(x) + \frac{\rho}{2} \|x - y\|^2 \Big\}. \tag{3.3}$$

The proximal point method carries out the iteration $x^{k+1} = \operatorname{prox}_F(x^k)$, $k = 1, 2, \ldots$ and is known to converge to a minimum of $F(\cdot)$, if a minimum exists [Roc76].

The main idea of the bundle method is to replace problem (3.1) with a sequence of approximate problems of the following form:

$$\min_x \tilde{F}^k(x) + \frac{\rho}{2} \|x - x^k\|^2. \tag{3.4}$$

Here $k = 1, 2, \ldots$ is the iteration number, x^k is the current best approximation to the solution, and $\tilde{F}^k(\cdot)$ is a piecewise linear convex lower approximation of the function $F(\cdot)$. Two versions of the method differ in the way this approximation is constructed.

3.2.1 The Version with Multiple Cuts

In the version with multiple cuts, the approximations $\tilde{F}^k(\cdot)$ are constructed as follows:

$$\tilde{F}^k(x) = \max_{j \in J_k} \big\{ F(z^j) + \langle g^j, x - z^j \rangle \big\},$$

with some previously generated points z^j and subgradients $g^j \in \partial F(z^j)$, $j \in J_k$, where $J_k \subseteq \{1, \ldots, k\}$. The points z^j are solutions of problems (3.4) at earlier iterations of the method.

Thus, problem (3.4) differs from (3.2) by the fact that the function $F(\cdot)$ is replaced by a cutting plane approximation. The other difference between the bundle method and the proximal point method is that the solution z^{k+1} of problem (3.4) is subject to a sufficient improvement test, which decides whether the next proximal center x^{k+1} should be set to z^{k+1} or remain unchanged.

Bundle Method with Multiple Cuts

Step 0: Set $k = 1$, $J_1 = \{1\}$, $z^1 = x^1$, and select $g^1 \in \partial F(z^1)$. Choose parameter $\beta \in (0, 1)$, and a stopping precision $\varepsilon > 0$.

Step 1: Find the solution z^{k+1} of subproblem (3.4).

Step 2: If

$$F(x^k) - \tilde{F}^k(z^{k+1}) \leq \varepsilon, \tag{3.5}$$

then stop; otherwise, continue.

Step 3: If

$$F(z^{k+1}) \leq F(x^k) - \beta\big(F(x^k) - \tilde{F}^k(z^{k+1})\big), \tag{3.6}$$

then set $x^{k+1} = z^{k+1}$ (*descent step*); otherwise set $x^{k+1} = x^k$ (*null step*).

Step 4: Select a set J_{k+1} so that

$$J_k \cup \{k+1\} \supseteq J_{k+1} \supseteq \{k+1\} \cup \big\{j \in J_k : F(z^j) + \langle g^j, z^{k+1} - z^j \rangle = \tilde{F}^k(z^{k+1})\big\}.$$

Increase k by 1 and go to Step 1.

3.2.2 The Version with Cut Aggregation

In the version with cut aggregation, as described in [Kiw83] and [Rus06, sec. 7.4.4], the approximations $\tilde{F}^k(\cdot)$ have only two pieces:

$$\tilde{F}^k(x) = \max\big\{\bar{F}^k(x), F(z^k) + \langle g^k, x - z^k \rangle\big\},$$

with the last generated point z^k and the corresponding subgradient $g^k \in \partial F(z^k)$. The function $\bar{F}^k(x)$ is a convex combination of affine minorants $F(z^j) + \langle g^j, x - z^j \rangle$, constructed at previously generated points z^j with subgradients $g^j \in \partial F(z^j)$, where $1 \leq j < k$. This function is updated at each iteration, as specified in Step 4 of the algorithm below.

Bundle Method with Cut Aggregation

Step 0: Set $k = 1$, $J_1 = \{1\}$, $z^1 = x^1$, and select $g^1 \in \partial F(z^1)$. Choose parameter $\beta \in (0, 1)$, and a stopping precision $\varepsilon > 0$.

Step 1: Find the solution z^{k+1} of subproblem (3.4).

Step 2: If

$$F(x^k) - \tilde{F}^k(z^{k+1}) \leq \varepsilon, \tag{3.7}$$

then stop; otherwise, continue.

Step 3: If

$$F(z^{k+1}) \leq F(x^k) - \beta\big(F(x^k) - \tilde{F}^k(z^{k+1})\big), \tag{3.8}$$

then set $x^{k+1} = z^{k+1}$ (*descent step*); otherwise set $x^{k+1} = x^k$ (*null step*).

Step 4: Define

$$\bar{F}^{k+1}(x) = \theta_k \bar{F}^k(x) + (1 - \theta_k)\big[F(z^k) + \langle g^k, x - z^k \rangle\big], \tag{3.9}$$

where $\theta_k \in [0, 1]$ is such that the gradient of $\bar{F}^{k+1}(\cdot)$ is equal to the subgradient of $\tilde{F}^k(\cdot)$ at z^{k+1} that satisfies the optimality conditions for problem (3.4). Increase k by 1 and go to Step 1.

3.2.3 Convergence

Convergence of the bundle method (in both versions) for convex functions is well-known.

Theorem 3.2.1. *Suppose* $\operatorname{Argmin} F \neq \emptyset$ *and* $\varepsilon = 0$. *Then a point* $x^* \in \operatorname{Argmin} F$ *exists, such that:*

$$\lim_{k \to \infty} x^k = \lim_{k \to \infty} z^k = x^*.$$

Proof. The proof of this result (in slightly different versions) can be found in numerous references, such as [Kiw85, Thm. 4.9], [HUL93, Thm. XV.3.2.4], or [Rus06, Thm. 7.16]. □

3.3 Auxiliary results

In this section, we collect several auxiliary results on the properties of the bundle method in the general case. They are either refined versions or direct quotations of results presented in [Rus06, sec. 7.4]. We consider both versions of the method in parallel, with the corresponding versions of the functions $\tilde{F}^k(\cdot)$. All the results hold true for both versions, because the analysis of the method with multiple cuts uses the version with cut aggregation anyway; in the proofs we explain the minor differences between the methods.

We first prove that if a null step occurs at iteration k, then the optimal objective function values of consecutive subproblems are increasing, and the gap is bounded below by a quantity dependent on

$$v_k = F(x^k) - \tilde{F}^k(z^{k+1}). \tag{3.10}$$

We define the optimal objective function values of subproblem (3.4) at iteration k as:

$$\eta^k = \tilde{F}^k(z^{k+1}) + \frac{\rho}{2}\|z^{k+1} - x^k\|^2. \tag{3.11}$$

Note that $x^{k+1} = x^k$ at a null step.

Since the point z^{k+1} is the optimal solution of (3.4) at iteration k, the vector

$$s^{k+1} = -\rho\big(z^{k+1} - x^k\big). \tag{3.12}$$

is the subgradient of $\tilde{F}^k(\cdot)$ at z^{k+1} that features in the optimality conditions. Consequently, the point z^{k+1} is also the unique minimum of the problem

$$\min_x \Big\{\tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle + \frac{\rho}{2}\|x - x^k\|^2\Big\}, \tag{3.13}$$

and the values of (3.11) and (3.13) coincide. In the method with cut aggregation, by the definition of θ_k in (3.9) and by (3.12), we have

$$\bar{F}^{k+1}(x) = \tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle.$$

The addition of a new cut at z^{k+1} and possible deletion of inactive cuts (in the method without cut aggregation), creates a function $\tilde{F}^{k+1}(\cdot)$, which satisfies the inequality

$$\tilde{F}^{k+1}(x) \geq \max\big(\tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle, F(z^{k+1}) + \langle g^{k+1}, x - z^{k+1}\rangle\big). \tag{3.14}$$

In the method with cut aggregation, exact equality in (3.14) is true, but we use the inequality "$\geq$" in further considerations. Since the test for a descent step is not satisfied, we have

$$\tilde{F}^{k+1}(z^{k+1}) = F(z^{k+1}) > \tilde{F}^k(z^{k+1}).$$

The solution z^{k+1} of problem (3.13) is unique, due to the strong convexity of the function being minimized there. Therefore, the optimal value of (3.13) must increase after replacing $\tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle$ with the right hand side of (3.14). The optimal value η^{k+1} of (3.4) at iteration $k+1$ is at least as large, due to (3.14).

The key issue is to bound the actual increment from η^k to η^{k+1} from below.

Lemma 3.3.1. *If a null step is made at iteration k, then*

$$\eta^{k+1} \geq \eta^k + \frac{1-\beta}{2}\bar{\mu}_k v_k, \tag{3.15}$$

where

$$\bar{\mu}_k = \min\left\{1, \frac{(1-\beta)\rho v_k}{\|s^{k+1} - g^{k+1}\|^2}\right\}. \tag{3.16}$$

Proof. Using (3.14), we can bound the optimal value of the subproblem (3.4) at iteration $k+1$ as follows:

$$\begin{aligned}
\eta^{k+1} &\geq \min_x \Big\{ \max\Big(\tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle, \\
&\qquad F(z^{k+1}) + \langle g^{k+1}, x - z^{k+1}\rangle\Big) + \frac{\rho}{2}\|x - x^k\|^2\Big\} \\
&\geq \min_x \Big\{(1-\mu)\Big(\tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle\Big) \\
&\quad + \mu\Big(F(z^{k+1}) + \langle g^{k+1}, x - z^{k+1}\rangle\Big) + \frac{\rho}{2}\|x - x^k\|^2\Big\},
\end{aligned} \tag{3.17}$$

with any value of the parameter $\mu \in [0,1]$. Define

$$\begin{aligned}
\hat{Q}_k(\mu) = \min_x \Big\{&(1-\mu)\Big(\tilde{F}^k(z^{k+1}) + \langle s^{k+1}, x - z^{k+1}\rangle\Big) \\
&+ \mu\Big(F(z^{k+1}) + \langle g^{k+1}, x - z^{k+1}\rangle\Big) + \frac{\rho}{2}\|x - x^k\|^2\Big\}.
\end{aligned} \tag{3.18}$$

Due to (3.13), $\hat{Q}_k(0) = \eta^k$. It follows from (3.17) that the difference between η^{k+1} and η^k can be bounded from below by the increase in the optimal value $\hat{Q}_k(\mu)$, when μ moves away from zero. That is,

$$\eta^{k+1} - \eta^k \geq \max_{\mu\in[0,1]} \hat{Q}_k(\mu) - \hat{Q}_k(0).$$

By direct calculation and with a view to (3.12), the minimizer on the right hand side of (3.18) is

$$\hat{x}(\mu) = z^{k+1} + \frac{\mu}{\rho}\big(s^{k+1} - g^{k+1}\big).$$

To obtain the derivative of $\hat{Q}_k(\cdot)$, we calculate the partial derivative of the right-hand side of (3.18) with respect to μ and then substitute $x = \hat{x}(\mu)$. We obtain

$$\begin{aligned}\hat{Q}_k'(\mu) &= F(z^{k+1}) - \tilde{F}^k(z^{k+1}) + \langle g^{k+1} - s^{k+1}, \hat{x}(\mu) - z^{k+1}\rangle \\ &= F(z^{k+1}) - \tilde{F}^k(z^{k+1}) - \frac{\mu}{\rho}\big\|s^{k+1} - g^{k+1}\big\|^2.\end{aligned}$$

Thus, for any value of $\mu_k \in [0,1]$,

$$\begin{aligned}\eta^{k+1} - \eta^k \geq \hat{Q}_k(\mu_k) - \hat{Q}_k(0) &= \int_0^{\mu_k} \hat{Q}_k'(\mu)\, d\mu \\ &= \mu_k\left(F(z^{k+1}) - \tilde{F}^k(z^{k+1}) - \frac{\mu_k}{2\rho}\big\|s^{k+1} - g^{k+1}\big\|^2\right).\end{aligned}$$

Define

$$\mu_k = \min\left\{1, \frac{\rho\big(F(z^{k+1}) - \tilde{F}^k(z^{k+1})\big)}{\|s^{k+1} - g^{k+1}\|^2}\right\}.$$

Clearly, $\mu_k \in [0,1]$. Substitution into the last displayed relation implies the inequality

$$\eta^{k+1} - \eta^k \geq \frac{\mu_k}{2}\big(F(z^{k+1}) - \tilde{F}^k(z^{k+1})\big). \tag{3.19}$$

If a null step occurs at iteration k, then the update step rule (3.8) is violated. Thus, $F(z^{k+1}) - \tilde{F}^k(z^{k+1}) > (1-\beta)v_k$. Using this in (3.19), we obtain

$$\eta^{k+1} - \eta^k \geq \frac{1-\beta}{2}\mu_k v_k.$$

Since $\mu_k \geq \bar{\mu}_k$, the postulated bound (3.15) follows. □

We recall a useful bound of the changes from η^k to η^{k+1} at descent steps.

Lemma 3.3.2. *If a descent step occurs at iteration k, then*

$$\eta^{k+1} - \eta^k \geq -\rho\big\|x^{k+1} - x^k\big\|^2 \geq \frac{1}{\beta}\big(F(x^{k+1}) - F(x^k)\big). \tag{3.20}$$

Proof. See [Rus06, (7.68)-(7.69)]. □

The following lemma relates the values of the optimal value of (3.4), η^k, and the value $\tilde{F}(z^{k+1})$ at the solution of (3.4).

Lemma 3.3.3. *At every iteration we have the inequality:*

$$F(x^k) - \eta^k \geq \frac{1}{2}\big[F(x^k) - \tilde{F}^k(z^{k+1})\big].$$

Proof. Consider the function

$$\Phi(\tau) = (1-\tau)F(x^k) + \tau\tilde{F}^k(z^{k+1}) + \frac{\rho}{2}\big\|(1-\tau)x^k + \tau z^{k+1} - x^k\big\|^2.$$

By construction, $\Phi(1) = \eta^k$, and, due to the convexity of $\tilde{F}^k(\cdot)$,

$$\Phi(\tau) \geq \tilde{F}^k\big((1-\tau)x^k + \tau z^{k+1}\big) + \frac{\rho}{2}\big\|(1-\tau)x^k + \tau z^{k+1} - x^k\big\|^2, \quad \tau \in [0,1]. \tag{3.21}$$

By the definition of z^{k+1}, the right hand side of (3.21) is minimized at $\tau = 1$. Therefore, $\Phi'(1) \leq 0$. Differentiating, we obtain the inequality

$$-F(x^k) + \tilde{F}^k(z^{k+1}) + \rho\big\|z^{k+1} - x^k\big\|^2 \leq 0.$$

This implies that

$$\begin{aligned}\eta^k &= \tilde{F}^k(z^{k+1}) + \frac{\rho}{2}\big\|z^{k+1} - x^k\big\|^2 \leq \tilde{F}^k(z^{k+1}) + \frac{1}{2}\big[F(x^k) - \tilde{F}^k(z^{k+1})\big] \\ &= \frac{1}{2}\big[F(x^k) + \tilde{F}^k(z^{k+1})\big].\end{aligned}$$

This is equivalent to the postulated inequality. □

Finally, we recall the following bound of the Moreau–Yosida regularization.

Lemma 3.3.4. *For any point $x \in \mathbb{R}^n$ we have*

$$F_\rho(x) \leq F(x) - \big\|x - x^*\big\|^2 \varphi\left(\frac{F(x) - F(x^*)}{\big\|x - x^*\big\|^2}\right), \tag{3.22}$$

where

$$\varphi(t) = \begin{cases} t^2 & \text{if } t \in [0,1], \\ -1 + 2t & \text{if } t \geq 1. \end{cases}$$

Proof. See [Rus06, Lem. 7.12]. □

3.4 Rate of Convergence

Our objective in this section is to derive a worst-case bound on the rate of convergence of the method. To this end, we assume that $\varepsilon > 0$ at Step 2 (inequality (3.5)) and we bound the number of iterations needed to achieve this accuracy.

We make a key assumption about strong convexity of the function $F(\cdot)$.

Assumption 3.4.1. *The function $F(\cdot)$ has a unique minimum point x^* and a constant $\alpha > 0$ exists, such that*

$$F(x) - F(x^*) \geq \alpha \|x - x^*\|^2,$$

for all $x \in \mathbb{R}^n$ with $F(x) \leq F(x^1)$.

We first show that stopping test of Step 2 guarantees the objective function accuracy of order ε.

Lemma 3.4.1. *Suppose Assumption 3.4.1 is satisfied. Then at every iteration k we have*

$$F(x^k) - F(x^*) \leq \frac{F(x^k) - \eta^k}{\min(\alpha, 1)}. \tag{3.23}$$

Proof. Since $\tilde{F}^k(\cdot) \leq F(\cdot)$, we have

$$F_\rho(x^k) = \min_x \left\{ F(x) + \frac{\rho}{2}\|x - x^k\|^2 \right\} \geq \min_x \left\{ \tilde{F}^k(x) + \frac{\rho}{2}\|x - x^k\|^2 \right\} = \eta^k. \tag{3.24}$$

Consider two cases.

Case 1: If $F(x^k) - F(x^*) \leq \|x^k - x^*\|^2$, then (3.22) with $x = x^k$ yields

$$F_\rho(x^k) \leq F(x^k) - \frac{\big(F(x^k) - F(x^*)\big)^2}{\|x^k - x^*\|^2}.$$

Combining this inequality with (3.24), we conclude that

$$\frac{\big(F(x^k) - F(x^*)\big)^2}{\|x^k - x^*\|^2} \leq F(x^k) - \eta^k.$$

Substitution of the denominator by the upper bound $(F(x^k) - F(x^*))/\alpha$ implies (3.23).

Case 2: $F(x^k) - F(x^*) > \|x^k - x^*\|^2$. Then (3.22) yields

$$F_\rho(x^k) \leq F(x^k) - 2\big(F(x^k) - F(x^*)\big) + \|x^k - x^*\|^2.$$

In view of (3.24), we obtain

$$2\big(F(x^k) - F(x^*)\big) - \big\|x^k - x^*\big\|^2 \leq F(x^k) - \eta^k,$$

which implies that $F(x^k) - F(x^*) \leq F(x^k) - \eta^k$ in this case. □

Corollary 3.4.1. *Suppose Assumption 3.4.1 is satisfied. If the stoping test* (3.5) *is satisfied at iteration k, then*

$$F(x^k) - F(x^*) \leq \frac{\varepsilon}{\min(\alpha, 1)}. \tag{3.25}$$

To bound the number of iterations of the method needed to achieve the prescribed accuracy we consider two issues. First, we prove linear rate of convergence between descent steps. Then, we bound the numbers of null steps between consecutive descent steps.

By employing the bound of Lemma 3.4.1, we can address the first issue.

Lemma 3.4.2. *Suppose Assumption 3.4.1 is satisfied. Then at every descent step k we have*

$$F(z^{k+1}) - F(x^*) \leq (1 - \bar{\alpha}\beta)\big(F(x^k) - F(x^*)\big), \tag{3.26}$$

where $\bar{\alpha} = \min(\alpha, 1)$.

Proof. It follows from the update rule (3.8) that

$$F(z^{k+1}) \leq (1 - \beta)F(x^k) + \beta\tilde{F}^k(z^{k+1}).$$

Since $\tilde{F}^k(z^{k+1}) \leq \eta^k$, Lemma 3.4.1 yields

$$F(x^k) - F(x^*) \leq \frac{1}{\bar{\alpha}}\big(F(x^k) - \tilde{F}^k(z^{k+1})\big).$$

Combining these inequalities and simplifying, we conclude that

$$\begin{aligned} F(z^{k+1}) &\leq (1 - \beta)F(x^k) + \beta\big(\bar{\alpha}F(x^*) - \bar{\alpha}F(x^k) + F(x^k)\big) \\ &= F(x^k) - \bar{\alpha}\beta\big(F(x^k) - F(x^*)\big). \end{aligned}$$

Subtraction of $F(x^*)$ from both sides yields the linear rate (3.26). □

We now pass to the second issue of deriving an upper bound on the number of null steps between two consecutive descent steps. To this end, we analyze the evolution of the gap $F(x^k) - \eta^k$.

It follows from [Rus06, (7.64)] that for all k

$$\left\|x^k - x^*\right\|^2 \leq \left\|x^1 - x^*\right\|^2 + \frac{2(1-\beta)}{\beta\rho}\big[F(x^1) - F(x^*)\big].$$

Thus, a uniform upper bound exists on the norm of the subgradients collected at points x^k. Therefore, a uniform upper bound exists on the distances $\|z^{k+1} - x^k\|$. Consequently, the subgradients collected at the points z^{k+1} are uniformly bounded as well, and the bound depends on the starting point only. Consequently, a constant M exists such that

$$\left\|s^{k+1} - g^{k+1}\right\|^2 \leq \rho M$$

at all null steps. With no loss of generality, we assume that $\varepsilon \leq M$.

Lemma 3.4.3. *If a null step occurs at iteration k, then*

$$F(x^k) - \eta^{k+1} \leq \gamma\big(F(x^k) - \eta^k\big), \tag{3.27}$$

where

$$\gamma = 1 - \frac{(1-\beta)^2\varepsilon}{2M}. \tag{3.28}$$

Proof. By Lemma 3.3.1, we have

$$F(x^k) - \eta^{k+1} \leq F(x^k) - \eta^k - \frac{1-\beta}{2}\bar{\mu}_k v_k. \tag{3.29}$$

On the other hand,

$$v_k = F(x^k) - \tilde{F}^k(z^{k+1}) = F(x^k) - \eta^k + \frac{\rho}{2}\left\|z^{k+1} - x^k\right\|^2 \geq F(x^k) - \eta^k. \tag{3.30}$$

Combining the last two inequalities, we conclude that

$$\begin{aligned} F(x^k) - \eta^{k+1} &\leq F(x^k) - \eta^k - \frac{1-\beta}{2}\bar{\mu}_k\big(F(x^k) - \eta^k\big) \\ &= \left(1 - \frac{1-\beta}{2}\bar{\mu}_k\right)\big(F(x^k) - \eta^k\big). \end{aligned} \tag{3.31}$$

Consider the definition (3.16) of $\bar{\mu}_k$ in Lemma 3.3.1. If $\bar{\mu}_k = 1$, then $(1 - \frac{1-\beta}{2}\bar{\mu}_k)$ is no greater than the bound (3.28), because $\varepsilon \leq M$. Otherwise, $\bar{\mu}_k$ is given by the second case in (3.16). Since the algorithm does not stop, we have $v_k > \varepsilon$, and thus

$$\bar{\mu}_k = \frac{(1-\beta)\rho v_k}{\|s^{k+1} - g^{k+1}\|^2} \geq \frac{(1-\beta)\varepsilon}{M}.$$

Substitution to (3.31) yields (3.28). □

Let $x^{(\ell-1)}, x^{(\ell)}, x^{(\ell+1)}$ be three consecutive proximal centers for $\ell \geq 2$ in the algorithm. We want to bound the number of iterations made with proximal center $x^{(\ell)}$. To this end, we bound two quantities: $F(x^{(\ell)}) - \eta^{k(\ell)}$, where $k(\ell)$ is the *first* step with proximal center $x^{(\ell)}$, and $F(x^{(\ell)}) - \eta^{k'(\ell)}$, where $k'(\ell)$ is the *last* step with proximal center $x^{(\ell)}$.

In the following we discuss each issue separately.

Recall that according to the algorithm, $x^{(\ell)}$ is the optimal solution of the last subproblem with proximal center $x^{(\ell-1)}$. Let $\eta^{k(\ell)-1}$ be the optimal objective value of the subproblem, that is,

$$\eta^{k(\ell)-1} = \tilde{F}^{k(\ell)-1}(x^{(\ell)}) + \frac{\rho}{2}\|x^{(\ell)} - x^{(\ell-1)}\|^2.$$

Lemma 3.4.4. *If a descent step is made at iteration $k(\ell) - 1$, then*

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq \frac{3}{2\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big). \tag{3.32}$$

Proof. The left inequality in (3.20) yields

$$\eta^{k(\ell)} \geq \eta^{k(\ell)-1} - \rho\|x^{(\ell)} - x^{(\ell-1)}\|^2.$$

Since $F(x^{(\ell)}) \leq F(x^{(\ell-1)})$, we obtain

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq F(x^{(\ell-1)}) - \eta^{k(\ell)-1} + \rho\|x^{(\ell)} - x^{(\ell-1)}\|^2.$$

As iteration $k(\ell) - 1$ is a descent step, the update rule (3.8) holds. Thus

$$\begin{aligned} F(x^{(\ell-1)}) - \eta^{k(\ell)-1} &= \left[F(x^{(\ell-1)}) - \tilde{F}^{k(\ell)-1}(x^{(\ell)})\right] - \frac{\rho}{2}\|x^{(\ell)} - x^{(\ell-1)}\|^2 \\ &\leq \frac{1}{\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big) - \frac{\rho}{2}\|x^{(\ell)} - x^{(\ell-1)}\|^2. \end{aligned}$$

Combining the last two inequalities we obtain

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq \frac{1}{\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big) + \frac{\rho}{2}\|x^{(\ell)} - x^{(\ell-1)}\|^2.$$

The right inequality in (3.20) can be now used to substitute $\left\|x^{(\ell)} - x^{(\ell-1)}\right\|^2$ on the right hand side to obtain (3.32). □

We can now integrate our results.

Applying Lemma 3.4.1, we obtain the following inequality at *every* null step with prox center $x^{(\ell)}$:

$$F(x^{(\ell)}) - \eta^k \geq \bar{\alpha}\big(F(x^{(\ell)}) - F(x^*)\big) \geq \bar{\alpha}\big(F(x^{(\ell)}) - F(x^{\ell+1})\big). \tag{3.33}$$

From Lemma 3.4.4 we know that for $2 \leq \ell < L$, where L is the last proximal center, the initial value of the left hand side (immediately after the previous descent step) is bounded from above by the expression on the right hand side of (3.32). Lemma 3.4.3 established a linear rate of decrease of the left hand side of (3.33). Therefore, the number n_ℓ of null steps with proximal center $x^{(\ell)}$, if it is positive, satisfies the inequality:

$$\frac{3}{2\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big)\gamma^{n_\ell - 1} \geq \bar{\alpha}\big(F(x^{(\ell)}) - F(x^{(\ell+1)})\big).$$

Consequently, for $2 \leq \ell < L$ we obtain the following upper bound on the number of null steps:

$$n_\ell \leq 1 + \frac{1}{\ln(\gamma)} \ln\left(\frac{2\beta\bar{\alpha}}{3}\frac{F(x^{(\ell)}) - F(x^{(\ell+1)})}{F(x^{(\ell-1)}) - F(x^{(\ell)})}\right). \tag{3.34}$$

If the number n_ℓ of null steps is zero, inequality (3.26) yields

$$\frac{F(x^{(\ell)}) - F(x^{(\ell+1)})}{F(x^{(\ell-1)}) - F(x^{(\ell)})} \leq \frac{F(x^{(\ell)}) - F(x^*)}{F(x^{(\ell-1)}) - F(x^*) - \big(F(x^{(\ell)}) - F(x^*)\big)} \leq \frac{1}{\frac{1}{1-\bar{\alpha}\beta} - 1}.$$

Elementary calculations then prove that both logarithms on the right hand side of (3.34) are negative, and thus inequality (3.34) is satisfied in this case as well.

Suppose there are L proximal centers appearing throughout the algorithm: $x^{(1)}$, $x^{(2)}$, $\ldots$, $x^{(L)}$. They divide the progress of the algorithm into L series of null steps. For the first series, similar to the analysis above, we use (3.33) and Lemma 3.4.3 to obtain the bound

$$n_1 \leq 1 + \frac{1}{\ln(\gamma)} \ln\left(\bar{\alpha}\frac{F(x^{(1)}) - F(x^{(2)})}{F(x^{(1)}) - \eta^1}\right).$$

For the last series, we use Lemma 3.3.3 to derive the inequality $F(x^{(\ell)}) - \eta^k \geq \varepsilon/2$, which must hold at every iteration at which the stopping test is not satisfied. We use it instead

of (3.33) in our analysis, and we obtain

$$n_L \leq 1 + \frac{1}{\ln(\gamma)} \ln\left(\frac{\beta}{3} \frac{\varepsilon}{F(x^{(L-1)}) - F(x^{(L)})}\right).$$

We aggregate the total number of null steps for different proximal centers and we obtain the following bound:

$$\sum_{\ell=1}^{L} n_\ell \leq \frac{1}{\ln(\gamma)} \left[\ln(\bar{\alpha}) + (L-2)\ln\left(\frac{2\beta\bar{\alpha}}{3}\right) + \ln\left(\frac{\beta}{3}\right) + \ln\left(\frac{\varepsilon}{F(x^1) - \eta^1}\right)\right] + L. \quad (3.35)$$

Let us recall the definition of γ in (3.28), and denote

$$C = \frac{(1-\beta)^2}{2M},$$

so that $\gamma = 1 - \varepsilon C$. Since $\ln(1 - \varepsilon C) < -\varepsilon C$, we derive the following inequality for the number of null steps:

$$\sum_{\ell=1}^{L} n_\ell \leq \frac{1}{-\varepsilon C} \left[\ln(\bar{\alpha}) + (L-2)\ln\left(\frac{2\beta\bar{\alpha}}{3}\right) + \ln\left(\frac{\beta}{3}\right) + \ln\left(\frac{\varepsilon}{F(x^1) - \eta^1}\right)\right] + L. \quad (3.36)$$

Let us now derive an upper bound on the number L of proximal points. By virtue of (3.5) and (3.8), descent steps are made only if

$$F(x^k) - F(x^*) \geq \beta\varepsilon;$$

otherwise, the method must stop. To explain it more specifically, if $F(x^k) - F(x^*) \leq \beta\varepsilon$, then $F(x^k) - F(z^{k+1}) \leq \beta\varepsilon$. If a descent step is made, $F(z^{k+1}) \leq F(x^k) - \beta v_k$. Then $\beta v_k \leq \beta\varepsilon$, $v_k \leq \varepsilon$. Thus we cannot make a descent step because the algorithm has already stopped, which contradicts our assumption. It follows from Lemma 3.4.2, that

$$(1 - \bar{\alpha}\beta)^{L-1}\big(F(x^1) - F(x^*)\big) \geq \beta\varepsilon.$$

Therefore,

$$L \leq 1 + \frac{\ln(\beta\varepsilon) - \ln\big(F(x^1) - F(x^*)\big)}{\ln(1 - \bar{\alpha}\beta)}. \quad (3.37)$$

As a result, we have the final bound for the total number of descent and null steps:

$$\begin{aligned} & L - 1 + \sum_{\ell=1}^{L} n_\ell \\ & \leq \frac{1}{\varepsilon C}\left[\frac{\ln\left(\frac{F(x^1) - F(x^*)(1-\bar{\alpha}\beta)}{\beta\varepsilon}\right)}{\ln(1-\bar{\alpha}\beta)} \ln\left(\frac{2\beta\bar{\alpha}}{3}\right) + \ln(\frac{1}{\bar{\alpha}}) + \ln\left(\frac{3}{\beta}\right)\right. \\ & \quad \left. \ln\left(\frac{F(x^1) - \eta^1}{\varepsilon}\right)\right] + 2\frac{\ln(\beta\varepsilon) - \ln\big(F(x^1) - F(x^*)\big)}{\ln(1-\bar{\alpha}\beta)} + 1. \end{aligned} \quad (3.38)$$

Therefore in order to achieve precision ε, the number of steps needed is of order

$$L + \sum_{\ell=1}^{L} n_\ell \sim \mathcal{O}\left(\frac{1}{\varepsilon} \ln\left(\frac{1}{\varepsilon}\right)\right).$$

This is almost equivalent to saying that given the number of iterations k, the precision of the solution is approximately $\mathcal{O}(1/k)$.

This chapter is published in [DR17].

Chapter 4

Selective Linearization for Multi-Block Convex Optimization

4.1 Introduction

In recent years, we have seen extensive development of the theory and methods for *structured regularization*, one of the most fundamental techniques to address the "big data" challenge. The basic problem is to minimize the following objective function with two components (blocks):

$$\min \Big[F(x) = f_1(x) + f_2(x) \Big], \tag{4.1}$$

where $f_1(\cdot)$ is the loss function and $f_2(\cdot)$ is a penalty function that imposes structured regularization to the model. Both functions are usually convex, but may be nonsmooth. Many data mining and machine learning problems can be cast within this framework, and efficient methods were proposed for these problems. The first group are the *operator splitting* methods originating from [DR56] and [PR55], and later developed and analyzed by [BC11, Com09, EB92, LM79], among others. Their dual versions, known as *Alternating Direction Methods of Multipliers* (ADMM) (see, [GM76, GM75, GT89]), found many applications in signal processing (see, *e.g.*, [BPC$^+$10, CP11], and the references therein). Sometimes, they are called *split Bregman methods* (see, *e.g.*, [GO09, YX11]).

The *Alternating Linearization Method* (ALIN) of [KRR99] handles problems of form (4.1) by introducing an additional improvement test to the operator splitting methods, which decides whether the proximal center should be updated or stay unchanged, and which of the operator splitting formulas should be applied at the current iteration. Its convergence mechanism is different than that of the splitting methods; it adapts some ideas of bundle methods of nonsmooth optimization [HUL93, Kiw85, Rus06]. The recent application of ALIN to structured regularization problems in [LPR14] proved very successful, with fast

convergence, good accuracy, and scalability to very large dimensions. In may be worth noticing that the recent application of the idea of alternating linearization by [GMS13] removes the update test from the method of [KRR99], thus effectively reducing it to an operator splitting method.

Most of existing techniques for structured regularization are designed to handle the two-block problem of form (4.1).

In this section, we plan to extend the ALIN framework to optimization problems involving multiple components. Namely, we aim to solve the following problem:

$$\min \Big\{ F(x) = \sum_{i=1}^{N} f_i(x) \Big\}, \tag{4.2}$$

with convex (possibly nondifferentiable) functions $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, N$, where the number of component functions, N, may be arbitrarily large. We only assume that the minimum exists.

In a typical application, $f_1(\cdot)$ may be the loss function, similar to problem (4.1), while the penalty function is a sum of multiple components. This type of generalization has many practical applications, including low rank matrix completion, compressed sensing, dynamic network analysis, and computer vision.

To the best of the authors' knowledge, It is known that a direct generalization of the ADMM to three or more blocks may fail to converge [CHYY14]. A known way is to introduce N copies $x^1 = x^2 = \cdots = x^N$ of x, and reduce the problem to the two-function case in the space $\mathbb{R}^{nN}$ [CP11]:

$$\min \sum_{i=1}^{N} f_i(x^i) + I(x^1, \ldots, x^N)$$

with $I(\cdot)$ denoting the indicator function of the subspace $x^1 = x^2 = \cdots = x^N$. Similar ideas were used in stochastic programming, under the name of *Progressive Hedging* [RW91]. A method for three blocks with one function being differentiable was theoretically analyzed in [CP12, Vu13].

Our new algorithm, which we call the *Selective Linearization Method* (SLIN), does not replicate the decision variables. It generates a sequence of points $x^k \in \mathbb{R}^n$ with a monotonic sequence of corresponding function values $\{F(x^k)\}$. At each iteration, it linearizes all but

one of the component functions and uses a proximal term penalizing for the distance to the last iterate. In a sense, each step is a backward step of the form employed in operator splitting. The order of processing the functions is not fixed; the method uses precise criteria for selecting the function to be treated exactly at the current step. It also employs special rules for updating the proximal center. These two rules distinguish our approach from the simultaneously proposed incremental proximal method of [Ber15], which applies to smooth functions only, and achieves linear convergence rate in this case.

The algorithm is a multi-block extension of the Alternating Linearization method for solving two-block nonsmooth optimization problems. Global convergence and convergence rate of the new algorithm are proved. Specifically, the new algorithm is proven to require at most $\mathcal{O}\big(\ln(1/\varepsilon)/\varepsilon\big)$ iterations to achieve solution accuracy ε. when the functions $f_i(\cdot)$ are smooth

In section 4.2, we present the method. And we prove its global convergence in section 4.3. The convergence rate is derived in section 4.4.

4.2 The SLIN Method

Our method derives from two fundamental ideas of convex optimization: the *Moreau–Yosida regularization* of $F(\cdot)$,

$$F_D(y) = \min\Big\{F(x) + \frac{1}{2}\big\|x-y\big\|_D^2\Big\}, \tag{4.3}$$

and the *proximal step* for (4.2),

$$\mathrm{prox}_F(y) = \arg\min\Big\{F(x) + \frac{1}{2}\big\|x-y\big\|_D^2\Big\}. \tag{4.4}$$

In the formulas above, the norm $\|x\|_D = \big(\langle x, Dx\rangle\big)^{1/2}$ with a positive definite matrix D. In applications, we shall use a diagonal D, which leads to major computational simplifications. The *proximal point method* carries out the iteration $x^{k+1} = \mathrm{prox}_F(x^k)$, $k = 1, 2, \ldots$ and is known to converge to a minimum of $F(\cdot)$, if a minimum exists [Roc76].

The main idea of our method is to replace problem (4.2) with a sequence of approximate problems of the following form:

$$\min_x f_{j_k}(x) + \sum_{i\neq j_k} \tilde{f}_i^k(x) + \frac{1}{2}\big\|x-x^k\big\|_D^2. \tag{4.5}$$

Here $k = 1, 2, \ldots$ is the iteration number, x^k is the current best approximation to the solution, $j_k \in \{1, \ldots, N\}$ is an index selected at iteration k, and $\tilde{f}_i^k$ are *affine minorants* of the functions f_i, $i \in \{1, \ldots, N\} \setminus \{j_k\}$. These minorants are constructed as follows:

$$\tilde{f}_i^k(x) = f_i(z_i^k) + \langle g_i^k, x - z_i^k \rangle,$$

with some points $z_i^k \in \mathbb{R}^n$ and specially selected subgradients $g_i^k \in \partial f_i(z_i^k)$. Thus, problem (4.5) differs from the proximal point problem in (4.3) by the fact that only one of the functions $f_i(\cdot)$ is treated exactly, while the other functions are replaced by affine approximations.

The key elements of the method are the selection of the index j_k, the way the affine approximations are constructed, and the update rule for the proximal center x^k. In formula (4.5) and in the algorithm description below we write simply $i \neq j_k$ for $i \in \{1, \ldots, N\} \setminus \{j_k\}$. We also write j in place of j_k; it will not lead to any misunderstanding.

We denote the function approximating $F(x)$ in (4.5) by

$$\tilde{F}^k(x) = f_{j_k}(x) + \sum_{i \neq j_k} \tilde{f}_i^k(x).$$

Selective Linearization (SLIN) Algorithm

Step 0: Set $k = 1$ and $j_1 \in \{1, \ldots, N\}$, select $x^1 \in \mathbb{R}^n$ and, for all $i \neq j_1$, linearization points $z_i^1 \in \mathbb{R}^n$ where the corresponding subgradients $g_i^1 \in \partial f_i(z_i^1)$ exist. Define $\tilde{f}_i^1(x) = f_i(z_i^1) + \langle g_i^1, x - z_i^1 \rangle$ for $i \neq j_1$. Choose parameters $\beta \in (0, 1)$, and a stopping precision $\varepsilon > 0$.

Step 1: Find the solution $z_{j_k}^k$ of the f_{j_k}-subproblem (4.5) and define

$$g_{j_k}^k = -\sum_{i \neq j_k} g_i^k - D(z_{j_k}^k - x^k). \tag{4.6}$$

Step 2: If

$$F(x^k) - \tilde{F}^k(z_{j_k}^k) \leq \varepsilon, \tag{4.7}$$

then stop. Otherwise, continue.

Step 3: If

$$F(z_{j_k}^k) \leq F(x^k) - \beta\big(F(x^k) - \tilde{F}^k(z_{j_k}^k)\big), \tag{4.8}$$

then set $x^{k+1} = z_{j_k}^k$ (*descent step*); otherwise set $x^{k+1} = x^k$ (*null step*).

Step 4: Select

$$j_{k+1} = \arg\max_{i \neq j_k} \{ f_i(z_{j_k}^k) - \tilde{f}_i^k(z_{j_k}^k) \}. \tag{4.9}$$

For all $i \neq j_{k+1}$, set $z_i^{k+1} = z_i^k$ and $g_i^{k+1} = g_i^k$ (so that $\tilde{f}_i^{k+1}(\cdot) \equiv \tilde{f}_i^k(\cdot)$). Increase k by 1 and go to Step 1.

Few comments are in order. Since the point $z_{j_k}^k$ is a solution of the subproblem (4.5), the vector $g_{j_k}^k$ calculated in (4.6) is indeed a subgradient of f_{j_k} at $z_{j_k}^k$; in fact, it is exactly the subgradient that features in the optimality condition for (4.5) at $z_{j_k}^k$. Therefore, at all iterations, the functions $\tilde{f}_i^k(\cdot)$ are minorants of the functions $f_i(\cdot)$. This in turn implies that $\tilde{F}^k(\cdot)$ is a lower approximation of $F(\cdot)$. Consequently, $F(x^k) - \tilde{F}^k(z_{j_k}^k) \geq 0$ in (4.7), with $F(x^k) = \tilde{F}^k(z_{j_k}^k)$ equivalent to x^k being the minimizer of $F(\cdot)$.

In practical implementation of the algorithm, the points z_i^k need not be stored. It is sufficient to memorize $\alpha_i^k = f_i(z_i^k) - \langle g_i^k, z_i^k \rangle$ and the subgradients g_i^k. At Step 4, we then set $\alpha_i^{k+1} = \alpha_i^k$ and $g_i^{k+1} = g_i^k$ for all $i \in \{1, \ldots, N\} \setminus \{j_k, j_{k+1}\}$, while $\alpha_{j_k}^{k+1} = f_i(z_i^k) - \langle g_i^k, z_i^k \rangle$. For j_{k+1} these data are not needed, because the function $f_{j_{k+1}}(\cdot)$ will not be linearized at the next iteration.

In some cases, the storage of the subgradients g_i^k may be substantially simplified.

Example 4.2.1. Suppose

$$F(x) = \sum_{i=1}^{N} \varphi_i(a_i^T x),$$

with convex functions $\varphi_i : \mathbb{R} \to \mathbb{R}$ and $a_i \in \mathbb{R}^n$, $i - 1, \ldots, n$. Then every subgradient of $f_i(x) = \varphi_i(a_i^T x)$ has the form $g_i^k = \sigma_i^k a_i$, with $\sigma_i^k \in \partial\varphi_i(a_i^T z_i^k)$. The scalars σ_i^k are sufficient for recovering the subgradients, because the vectors a_i are part of the problem data.

4.3 Global convergence

We assume that $\varepsilon = 0$ in Step 2. To prove convergence of the algorithm, we consider two cases: with finitely or infinitely many descent steps.

We first address the finite case and show that the proximal center updated in the last descent step must be an optimal solution to problem (4.2). To this end, we prove that if a null step is made at iteration k, then the optimal objective function values of consecutive

subproblems are increasing and the gap is bounded below by a value determined by

$$v_k = F(x^k) - \tilde{F}^k(z^k_{j_k}). \tag{4.10}$$

We shall also use this result in the proof of convergence rate.

We denote the optimal objective function value of subproblem (4.5) at iteration k by

$$\eta^k = \min_x f_{j_k}(x) + \sum_{i \neq j_k} \tilde{f}^k_i(x) + \frac{1}{2}\|x - x^k\|^2_D.$$

Lemma 4.3.1. *If a null step is made at iteration k, then*

$$\eta^{k+1} \geq \eta^k + \frac{1-\beta}{2(N-1)}\bar{\mu}_k v_k, \tag{4.11}$$

where

$$\bar{\mu}_k = \min\Big\{1, \frac{(1-\beta)v_k}{(N-1)\|s^k_{j_{k+1}} - g^k_{j_{k+1}}\|^2_{D^{-1}}}\Big\}, \tag{4.12}$$

with an arbitrary $s^k_{j_{k+1}} \in \partial f_{j_{k+1}}(z^k_{j_k})$.

Proof. The change from the f_{j_k}-subproblem to the $f_{j_{k+1}}$-subproblem can be viewed as two steps: first is the change of $f_{j_k}(\cdot)$ to $\tilde{f}^k_{j_k}(\cdot)$, followed by the change of $\tilde{f}^k_{j_{k+1}}(\cdot)$ to $f_{j_{k+1}}(\cdot)$. By the selection of the subgradient (4.6) and the resulting construction of $\tilde{f}^k_{j_k}(\cdot)$, the first operation does not change the solution and the optimal value of the subproblem. Thus the optimal value of (4.5) satisfies the following equation:

$$\eta^k = \min_x \sum_{i=1}^{N} \tilde{f}^k_i(x) + \frac{1}{2}\|x - x^k\|^2_D. \tag{4.13}$$

Since $x^{k+1} = x^k$ at a null step, and $f_{j_{k+1}} \geq \tilde{f}^k_{j_{k+1}}$, the second operation can only increase the optimal value of the last problem. Therefore, $\eta^{k+1} \geq \eta^k$.

Consider the family of relaxations of the $f_{j_{k+1}}$-subproblem at iteration $k+1$:

$$\begin{aligned}\hat{Q}_k(\mu) = \min_x \Big\{ &\sum_{i \neq j_{k+1}} \tilde{f}^{k+1}_i(x) + (1-\mu)\Big(f_{j_{k+1}}(z^k_{j_{k+1}}) + \langle g^k_{j_{k+1}}, x - z^k_{j_{k+1}}\rangle\Big) \\ &+ \mu\Big(f_{j_{k+1}}(z^k_{j_k}) + \langle s^k_{j_{k+1}}, x - z^k_{j_k}\rangle\Big) + \frac{1}{2}\|x - x^k\|^2_D\Big\},\end{aligned} \tag{4.14}$$

with parameter $\mu \in [0,1]$. In the above relaxation, the function $f_{j_{k+1}}(\cdot)$ is replaced by a convex combination of its two affine minorants: one at the point $z^k_{j_{k+1}}$, which is $\tilde{f}^k_{j_{k+1}}(\cdot)$ used at iteration k, and the other one at the kth trial point $z^k_{j_k}$, with an arbitrary subgradient

$s^k_{j_{k+1}}$. Due to (4.13), the value of (4.14) with $\mu = 0$ coincides with η^k. Therefore, the difference between η^{k+1} and η^k can be estimated from below by the increase in the optimal value $\hat{Q}_k(\mu)$ of (4.14) when μ moves away from zero. That is,

$$\eta^{k+1} - \eta^k \geq \max_{\mu\in[0,1]} \hat{Q}_k(\mu) - \hat{Q}_k(0). \tag{4.15}$$

Define $\delta_k = F(z^k_{j_k}) - \tilde{F}^k(z^k_{j_k})$. Note that $\delta_k \geq 0$, since $f_i \geq \tilde{f}^k_i$ for $i \neq j_k$. We also define $\mu_k = \min\Big\{1, \frac{\delta_k}{(N-1)\big\|s^k_{j_{k+1}} - g^k_{j_{k+1}}\big\|^2_{D^{-1}}}\Big\}$, so $\mu_k \in [0,1]$.

Here we use the fact of parametric optimization (see Danskin's theorem in [Ber03]). By direct calculation, and with the use of (4.6), the solution of (4.14) has the form

$$\hat{x}(\mu) = x^k - D^{-1}\Big[\sum_{i=1}^N g^k_i + \mu\big(s^k_{j_{k+1}} - g^k_{j_{k+1}}\big)\Big] = z^k_{j_k} - \mu D^{-1}\big(s^k_{j_{k+1}} - g^k_{j_{k+1}}\big).$$

Using the definitions following (4.14) and the fact that $\hat{x}(0) = z^k_{j_k}$, the derivative of $\hat{Q}_k$ can be expressed as first differentiate with respect to μ and then substitute $\hat{x}(\mu)$ to the derivative as follows:

$$\begin{aligned}
\hat{Q}'_k(\mu) &= \langle s^k_{j_{k+1}} - g^k_{j_{k+1}}, \hat{x}(\mu)\rangle \\
&\qquad + \Big(f_{j_{k+1}}(z^k_{j_k}) - \langle s^k_{j_{k+1}}, z^k_{j_k}\rangle\Big) - \Big(f_{j_{k+1}}(z^k_{j_{k+1}}) - \langle g^k_{j_{k+1}}, z^k_{j_{k+1}}\rangle\Big) \\
&= \langle s^k_{j_{k+1}} - g^k_{j_{k+1}}, \hat{x}(\mu) - z^k_{j_k}\rangle \\
&\qquad + f_{j_{k+1}}(z^k_{j_k}) - \Big(f_{j_{k+1}}(z^k_{j_{k+1}}) + \langle g^k_{j_{k+1}}, z^k_{j_k} - z^k_{j_{k+1}}\rangle\Big) \\
&\geq \langle s^k_{j_{k+1}} - g^k_{j_{k+1}}, \hat{x}(\mu) - z^k_{j_k}\rangle + \frac{F(z^k_{j_k}) - \tilde{F}^k(z^k_{j_k})}{N-1} \\
&= -\mu\big\|s^k_{j_{k+1}} - g^k_{j_{k+1}}\big\|^2_{D^{-1}} + \frac{\delta_k}{N-1}.
\end{aligned} \tag{4.16}$$

In the inequality above, we used the definition (4.9) of j_{k+1} and the fact that the maximum of the differences $f_j(z^k_{j_k}) - \tilde{f}^k_j(z^k_{j_k})$ over $j \neq j_k$ is larger than their average. Thus

$$\hat{Q}_k(\mu_k) - \hat{Q}_k(0) = \int_0^{\mu_k} \hat{Q}'_k(\mu)\, d\mu \geq \mu_k\left(\frac{\delta_k}{N-1} - \frac{1}{2}\mu_k\big\|s^k_{j_{k+1}} - g^k_{j_{k+1}}\big\|^2_{D^{-1}}\right). \tag{4.17}$$

Substitution of the definition of μ_k yields

$$\eta^{k+1} \geq \eta^k + \frac{\mu_k\delta_k}{2(N-1)}. \tag{4.18}$$

If a null step is made at iteration k, then the update step rule (4.8) is violated. Thus, $\delta_k = F(z^k_{j_k}) - \tilde{F}^k(z^k_{j_k}) > (1-\beta)v_k$. Plugging this lower bound on δ_k into (4.18) and using the definition of $\bar{\mu}_k$, we obtain the postulated bound (4.11).

Finally, we remark that $s^k_{j_{k+1}} \neq g^k_{j_{k+1}}$, because $f_{j_{k+1}}(z^k_{j_k}) > \tilde{f}^k_{j_{k+1}}(z^k_{j_k})$. □

We also need to estimate the size of the steps made by the method.

Lemma 4.3.2. *At every iteration k,*

$$\frac{1}{2}\big\|z^k_{j_k} - \operatorname{prox}_F(x^k)\big\|^2_D \leq F_D(x^k) - \eta^k. \tag{4.19}$$

Proof. Since $F(\cdot) \geq \tilde{F}^k(\cdot)$ and $z^k_{j_k}$ is a solution of the strongly convex problem (4.5), we have

$$\begin{aligned} F_D(x^k) &= F\big(\operatorname{prox}_F(x^k)\big) + \frac{1}{2}\big\|\operatorname{prox}_F(x^k) - x^k\big\|^2_D \\ &\geq \tilde{F}^k\big(\operatorname{prox}_F(x^k)\big) + \frac{1}{2}\big\|\operatorname{prox}_F(x^k) - x^k\big\|^2_D \\ &\geq \tilde{F}^k(z^k_{j_k}) + \frac{1}{2}\big\|z^k_{j_k} - x^k\big\|^2_D + \frac{1}{2}\big\|z^k_{j_k} - \operatorname{prox}_F(x^k)\big\|^2_D \\ &= \eta^k + \frac{1}{2}\big\|z^k_{j_k} - \operatorname{prox}_F(x^k)\big\|^2_D. \end{aligned} \tag{4.20}$$

Rearranging, we obtain (4.19). □

We are now ready to prove optimality in the case of finitely many descent steps.

Theorem 4.3.1. *Suppose $\varepsilon = 0$, the set $\mathcal{K} = \{1\} \cup \{k > 1 : x^k \neq x^{k-1}\}$ is finite and $\inf F > -\infty$. Let $m \in \mathcal{K}$ be the largest index such that $x^m \neq x^{m-1}$. Then $x^m \in \operatorname{Argmin} F$.*

Proof. We argue by contradiction. Suppose $x^m \notin \operatorname{Argmin} F$. If $\varepsilon = 0$ the method cannot stop, because $\tilde{F}^k(z^k_{j_k}) \leq F(\operatorname{prox}_F(x^m)) < F(x^m)$, for all $k \geq m$. Therefore, null steps are made at all iterations $k \geq m$, with $x^k = x^m$. By Lemma 4.3.1, the sequence $\{\eta^k\}$ is nondecreasing and bounded above by $F(x^m)$. Hence $\eta^{k+1} - \eta^k \to 0$. The right hand side of estimate (4.19) with $x^k = x^m$ for $k \geq m$, owing to the monotonicity of $\{\eta^k\}$, is nonincreasing, and thus the sequence $\{z^k_{j_k}\}$ is bounded. Since the subgradients of a finite-valued convex function are locally bounded (see Theorem 23.4 in [Roc70]), the differences $\big\|s^k_{j_{k+1}} - g^k_{j_{k+1}}\big\|_{D^{-1}}$ appearing in the definition of $\bar{\mu}_k$ in Lemma 4.3.1 are bounded from above. Therefore, $v_k \to 0$. As $F(x^m) \geq \eta^k \geq F(x^m) - v_k$, we have $\eta^k \uparrow F(x^m)$.

On the other hand, the inequality $\tilde{F}^k(\cdot) \leq F(\cdot)$ implies that $\eta^k \leq F_D(x^m)$ for all $k \geq m$. Since $x^m \notin \operatorname{Argmin} F$, we have $F_D(x^m) < F(x^m)$, which contradicts the convergence of $\{\eta^k\}$ to $F(x^m)$. □

We now address the infinite case. Note that the update test (4.8) can be expressed as follows:

$$\tilde{F}^k(z^k_{j_k}) \geq -\frac{1}{\beta}F(z^k_{j_k}) + \frac{1-\beta}{\beta}F(x^k). \tag{4.21}$$

Theorem 4.3.2. *Suppose* $\operatorname{Argmin} F \neq \emptyset$. *If the set* $\mathcal{K} = \{k : x^{k+1} \neq x^k\}$ *is infinite, then* $\lim_{k\to\infty} x^k = x^*$, *for some* $x^* \in \operatorname{Argmin} F$.

Proof. Consider iteration $k \in \mathcal{K}$ (descent step). From the optimality condition for (4.5) we obtain

$$0 \in \partial\Big[f_{j_k}(z^k_{j_k}) + \sum_{i\neq j_k} \tilde{f}^k_i(z^k_{j_k})\Big] + D\big(z^k_{j_k} - x^k\big), \tag{4.22}$$

which yields

$$D(x^k - x^{k+1}) \in \partial\Big[f_{j_k}(z^k_{j_k}) + \sum_{i\neq j_k} \tilde{f}^k_i(z^k_{j_k})\Big]. \tag{4.23}$$

Then for any point $x^* \in \operatorname{Argmin} F$ we obtain

$$F(x^*) \geq \tilde{F}^k(x^*) \geq \tilde{F}^k(x^{k+1}) + \big\langle D(x^k - x^{k+1}), x^* - x^{k+1}\big\rangle. \tag{4.24}$$

Hence

$$\begin{aligned}
\big\|x^{k+1} - x^*\big\|^2_D &= \big\|x^k - x^*\big\|^2_D + 2\big\langle D(x^{k+1} - x^k), x^k - x^*\big\rangle + \big\|x^{k+1} - x^k\big\|^2_D \\
&\leq \big\|x^k - x^*\big\|^2_D + 2\big\langle D(x^{k+1} - x^k), x^{k+1} - x^*\big\rangle \\
&\leq \big\|x^k - x^*\big\|^2_D + 2\big(F(x^*) - \tilde{F}^k(x^{k+1})\big).
\end{aligned}$$

Using (4.21), we can continue this chain of inequalities as follows

$$\begin{aligned}
\big\|x^{k+1} - x^*\big\|^2_D &\leq \big\|x^k - x^*\big\|^2_D + 2\big(F(x^*) - \frac{1}{\beta}F(x^{k+1}) + \frac{1-\beta}{\beta}F(x^k)\big) \\
&= \big\|x^k - x^*\big\|^2_D + 2\big(F(x^*) - F(x^k)\big) + \frac{2}{\beta}\big(F(x^k) - F(x^{k+1})\big).
\end{aligned} \tag{4.25}$$

Thus, adding up (4.25) for all $k \in \mathcal{K}$, $k \leq m$, and noting that the null steps do not change the proximal centers, we obtain

$$\big\|x^{m+1} - x^*\big\|^2_D \leq \big\|x^1 - x^*\big\|^2_D + 2\sum_{\substack{k\in\mathcal{K}\\ k\leq m}} \big(F(x^*) - F(x^k)\big) + \frac{2}{\beta}\sum_{\substack{k\in\mathcal{K}\\ k\leq m}} \big(F(x^k) - F(x^{k+1})\big). \tag{4.26}$$

The term $2\sum_{k\in\mathcal{K},k\leq m}\big(F(x^*) - F(x^k)\big)$ is non-positive, and the last term is bounded by $\frac{2}{\beta}\big(F(x^1) - F(x^*)\big)$. Thus, several conclusions follow from inequality (4.26). First, the sequence $\{x^k\}_{k\in\mathcal{K}}$ is bounded, because their distances to x^* are bounded. Secondly, rewriting

(4.26) as

$$\sum_{k\in\mathcal{K},k\le m}\big(F(x^k)-F(x^*)\big)\le\frac{1}{2}\big(\|x^1-x^*\|_D^2-\|x^{m+1}-x^*\|_D^2\big)+\frac{1}{\beta}\big(F(x^1)-F(x^{m+1})\big),$$

and letting $m\to\infty$ in $\mathcal{K}$, we deduce that

$$\sum_{k\in\mathcal{K}}\Big(F(x^k)-F(x^*)\Big)\le\frac{1}{2}\|x^1-x^*\|_D^2+\frac{1}{\beta}\big(F(x^1)-F(x^*)\big). \tag{4.27}$$

Consequently, $F(x^k)\to F(x^*)$ as $k\to\infty$ in $\mathcal{K}$. As the null steps do not change the proximal centers, we also have $F(x^k)\to F(x^*)$, when $k\to\infty$.

To prove that the sequence of proximal centers converges to an optimal solution, note that since the infinite sequence $\{x^k\}_{k\in\mathcal{K}}$ is bounded, it has a convergent subsequence whose limit $\hat{x}$ is a minimizer of F. Without loss of generality, we substitute $\hat{x}$ for x^* in the above derivations, and add (4.25) for all $k\in\mathcal{K}$ such that $\ell\le k\le m$. For any $1\le\ell\le m$ we obtain the following analog of (4.26):

$$\begin{aligned}\|x^{m+1}-\hat{x}\|_D^2&\le\|x^\ell-\hat{x}\|_D^2+2\sum_{\substack{k\in\mathcal{K}\\k\le m}}\big(F(\hat{x})-F(x^k)\big)+\frac{2}{\beta}\sum_{\substack{k\in\mathcal{K}\\k\le m}}\big(F(x^k)-F(x^{k+1})\big)\\&\le\|x^\ell-\hat{x}\|_D^2+\frac{2}{\beta}\big(F(x^\ell)-F(\hat{x})\big).\end{aligned}$$

The right hand side of the last inequality can be made arbitrarily small by choosing ℓ from the subsequence converging to $\hat{x}$. Therefore the entire sequence $\{x^k\}_{k\in\mathcal{K}}$ is convergent to $\hat{x}$. □

We finish this section with a number of conclusions, which will be useful in the analysis of the rate of convergence.

Lemma 4.3.3. *If there is a descent step at iteration k, then*

$$\eta^{k+1}-\eta^k\ge-\|x^{k+1}-x^k\|_D^2\ge\frac{1}{\beta}\big(F(x^{k+1})-F(x^k)\big). \tag{4.28}$$

Proof. By (4.6),

$$\sum_{i=1}^N g_i^k+D(x^{k+1}-x^k)=0. \tag{4.29}$$

The optimal value of (4.5) at iteration $k+1$ can be then estimated as follows:

$$\begin{aligned}\eta^{k+1} &= \min_x \left\{ f_{j_{k+1}}(x) + \sum_{i \neq j_{k+1}} \tilde{f}_i^{k+1}(x) + \frac{1}{2}\|x - x^{k+1}\|_D^2 \right\} \\ &\geq \min_x \left\{ \sum_{i=1}^N \tilde{f}_i^{k+1}(x) + \frac{1}{2}\|x - x^{k+1}\|_D^2 \right\} \\ &= \sum_{i=1}^N \tilde{f}_i^{k+1}(x^{k+1}) + \min_x \left\{ \Big\langle \sum_{i=1}^N g_i^k, x - x^{k+1} \Big\rangle + \frac{1}{2}\|x - x^{k+1}\|_D^2 \right\} \\ &= \tilde{F}^k(x^{k+1}) + \min_x \left\{ -\big\langle D(x^{k+1} - x^k), x - x^{k+1} \big\rangle + \frac{1}{2}\|x - x^{k+1}\|_D^2 \right\}.\end{aligned}$$

The minimizer on the right hand side is $x = 2x^{k+1} - x^k$, and we conclude that

$$\eta^{k+1} \geq \tilde{F}^k(x^{k+1}) - \frac{1}{2}\|x^{k+1} - x^k\|_D^2 = \eta^k - \|x^{k+1} - x^k\|_D^2,$$

which proves the left inequality in (4.28). To prove the right inequality, we observe that the test (4.8) for the descent step is satisfied at iteration k, and thus

$$F(x^k) - F(x^{k+1}) \geq \beta\big(F(x^k) - \tilde{F}^k(x^{k+1})\big) \geq \beta\big(\tilde{F}^k(x^k) - \tilde{F}^k(x^{k+1})\big).$$

The expression on the right hand side can be calculated with the use of (4.29), exactly as in the derivations above, which yields

$$F(x^k) - F(x^{k+1}) \geq \beta\|x^{k+1} - x^k\|_D^2.$$

This proves the right inequality in (4.28). □

We can now summarize convergence properties of the sequences generated by the algorithm.

Corollary 4.3.1. *Suppose* $\operatorname{Argmin} F \neq \emptyset$ *and* $\varepsilon = 0$. *Then a point* $x^* \in \operatorname{Argmin} F$ *exists, such that:*

(i) $\lim_{k\to\infty} x^k = \lim_{k\to\infty} z_{j_k}^k = x^*$;
(ii) $\lim_{k\to\infty} \eta^k = F(x^*)$.

Proof. The convergence of $\{x^k\}$ to a minimum point x^* has been proved in Theorems 4.3.1 and 4.3.2. It remains to verify the convergence properties of $\{z_{j_k}^k\}$ and $\{\eta^k\}$. It follows from Lemmas 4.3.1 and 4.3.3 that the sequence $\eta^k - \frac{1}{\beta}F(x^k)$ is nondecreasing. Since $\eta^k \leq F(x^k)$

by construction, the sequence η^k is bounded from above, and thus convergent. Therefore, a limit η^* of $\{\eta^k\}$ exists and $\eta^* \leq F(x^*)$. If the number of descent steps is finite, the equality $\eta^* = F(x^*)$ follows from Theorem 4.3.1. If the number of descent steps is infinite, inequality (4.8) at each descent step k yields:

$$F(x^k) - \eta^k \leq F(x^k) - \tilde{F}^k(x^{k+1}) \leq \frac{1}{\beta}\big(F(x^k) - F(x^{k+1})\big).$$

Passing to the limit over descent steps $k \to \infty$ we conclude that $\eta^* \geq F(x^*)$. Consequently, $\eta^* = F(x^*)$ and assertion (ii) is true.

The convergence of the sequence $\{z_{j_k}^k\}$ to x^* follows from inequality (4.19), because $x^k \to x^*$ and $\eta^k \to F(x^*)$. □

4.4 Rate of Convergence

Our objective in this section is to estimate the rate of convergence of the method. To this end, we assume that $\varepsilon > 0$ at Step 2 (inequality (4.7)) and we estimate the number of iterations needed to achieve this accuracy. We also make an additional assumption about the growth rate of the function $F(\cdot)$.

Assumption 4.4.1. *The function $F(\cdot)$ has a unique minimum point x^* and a constant $\alpha > 0$ exists, such that*

$$F(x) - F(x^*) \geq \alpha\big\|x - x^*\big\|_D^2,$$

for all $x \in \mathbb{R}^n$.

Assumption 4.4.1 has a number of implications on the properties of the method. First, we recall from [Rus06, Lem. 7.12] the following estimate of the Moreau–Yosida regularization.

Lemma 4.4.1. *For any point $x \in \mathbb{R}^n$, we have*

$$F_D(x) \leq F(x) - \big\|x - x^*\big\|_D^2\, \varphi\left(\frac{F(x) - F(x^*)}{\big\|x - x^*\big\|_D^2}\right), \tag{4.30}$$

where

$$\varphi(t) = \begin{cases} t^2 & \text{if } t \in [0,1], \\ -1 + 2t & \text{if } t \geq 1. \end{cases}$$

Proof. See [Rus06, Lem. 7.12]. □

Lemma 4.4.2. *Suppose Assumption 4.4.1 is satisfied. Then the stopping test* (4.7) *implies that*

$$F(x^k) - F(x^*) \leq \frac{\varepsilon}{\min(\alpha, 1)}. \tag{4.31}$$

Proof. As $\tilde{F}^k(\cdot) \leq F(\cdot)$, the stopping criterion implies that

$$\begin{aligned} F_D(x^k) = \min_x \Big\{ F(x) + \frac{1}{2}\big\|x - x^k\big\|_D^2 \Big\} &\geq \min_x \Big\{ \tilde{F}^k(x) + \frac{1}{2}\big\|x - x^k\big\|_D^2 \Big\} \\ &= \tilde{F}^k(z_{j_k}^k) + \frac{1}{2}\big\|z_{j_k}^k - x^k\big\|_D^2 \geq F(x^k) - \varepsilon. \end{aligned} \tag{4.32}$$

Consider two cases.

Case 1: If $F(x^k) - F(x^*) \leq \big\|x^k - x^*\big\|_D^2$, then (4.30) with $x = x^k$ yields

$$F_D(x^k) \leq F(x^k) - \frac{\big(F(x^k) - F(x^*)\big)^2}{\big\|x^k - x^*\big\|_D^2}.$$

Combining this inequality with (4.32), we conclude that

$$\frac{\big(F(x^k) - F(x^*)\big)^2}{\big\|x^k - x^*\big\|_D^2} \leq \varepsilon. \tag{4.33}$$

Substitution of the denominator by the upper estimate $(F(x^k) - F(x^*))/\alpha$ implies (4.31).

Case 2: $F(x^k) - F(x^*) > \big\|x^k - x^*\big\|_D^2$. Then (4.30) yields

$$F_D(x^k) \leq F(x^k) - 2\big(F(x^k) - F(x^*)\big) + \big\|x^k - x^*\big\|_D^2.$$

With a view to (4.32), we obtain

$$2\big(F(x^k) - F(x^*)\big) - \big\|x^k - x^*\big\|_D^2 \leq \varepsilon,$$

which implies that $F(x^k) - F(x^*) \leq \varepsilon$ in this case. □

Lemma 4.4.3. *Suppose Assumption 4.4.1 is satisfied. Then at any iteration k we have*

$$F(x^k) - \eta^k \geq \frac{2\varphi(\alpha)}{1 + 2\varphi(\alpha)}\big(F(x^k) - F(x^*)\big).$$

Proof. By Lemma 4.3.2,

$$F(x^k) - \eta^k \geq F(x^k) - F_D(x^k).$$

To derive a lower bound for the right hand side of the last inequality, we use Assumption 4.4.1 in (4.30) with $x = x^k$. We obtain

$$F_D(x^k) \le F(x^k) - \|x^k - x^*\|_D^2 \, \varphi(\alpha). \tag{4.34}$$

By the definition of the Moreau–Yosida regularization, for any optimal solution x^* we have

$$F(x^*) + \frac{1}{2}\|x^* - x^k\|_D^2 \ge F_D(x^k),$$

and thus

$$\|x^k - x^*\|_D^2 \ge 2\big(F_D(x^k) - F(x^*)\big).$$

Substitution to (4.34) yields

$$F(x^k) - F_D(x^k) \ge 2\big(F_D(x^k) - F(x^*)\big)\varphi(\alpha),$$

which can be manipulated to

$$F(x^k) - F_D(x^k) \ge \frac{2\varphi(\alpha)}{1 + 2\varphi(\alpha)}\big(F(x^k) - F(x^*)\big).$$

This can be combined with the first inequality in the proof, to obtain the desired result. □

In order to estimate the number of iterations of the method needed to achieve the prescribed accuracy, we need to consider two aspects. First, we prove linear rate of convergence between descent steps. Then, we estimate the numbers of null steps between consecutive descent steps.

By employing the estimate of Lemma 4.4.2, we can address the first aspect. To simplify notation, with no loss of generality, we assume that $\alpha \in (0, 1]$ (otherwise, we would have to replace α with $\bar{\alpha} = \min(\alpha, 1)$ in the considerations below).

Lemma 4.4.4. *Suppose x^* is the unique minimum point of $F(\cdot)$ and Assumption 4.4.1 is satisfied. Then at every descent step k, when the update step rule* (4.8) *is satisfied, we have the inequality:*

$$F(z_{j_k}^k) - F(x^*) \le (1 - \alpha\beta)\big(F(x^k) - F(x^*)\big). \tag{4.35}$$

Proof. It follows from the update rule (4.8) that

$$F(z_{j_k}^k) \le F(x^k) - \beta\big(F(x^k) - \tilde{F}^k(z_{j_k}^k)\big).$$

Using Lemma 4.4.2 with $\varepsilon = F(x^k) - \tilde{F}^k(z^k_{j_k})$, we obtain

$$F(x^k) - F(x^*) \le \frac{1}{\alpha}\big(F(x^k) - \tilde{F}^k(z^k_{j_k})\big).$$

Combining these inequalities and simplifying, we conclude that

$$\begin{aligned} F(z^k_{j_k}) &\le (1-\beta)F(x^k) + \beta\big(\alpha F(x^*) - \alpha F(x^k) + F(x^k)\big) \\ &= F(x^k) - \alpha\beta\big(F(x^k) - F(x^*)\big). \end{aligned}$$

Subtracting $F(x^*)$ from both sides, we obtain the linear rate (4.35). □

We now pass to the second issue: the estimation of the number of null steps between two consecutive descent steps. We shall base it on the analysis of the gap $F(x^k) - \eta^k$.

By virtue of Corollary 4.3.1, the sequence points $\{z^k_{j_k}\}$ generated by the algorithm are uniformly bounded. Since subgradients of finite-valued convex functions are locally bounded, the subgradients of all f_{j_k} are bounded, and thus a constant M exists, such that

$$\big\|s^k_{j_{k+1}} - g^k_{j_{k+1}}\big\|^2_{D^{-1}} \le M$$

at all null steps. With no loss of generality, we assume that $\varepsilon \le (N-1)M$.

Lemma 4.4.5. *If a null step is made at iteration k, then*

$$F(x^k) - \eta^{k+1} \le \gamma\big(F(x^k) - \eta^k\big), \tag{4.36}$$

where

$$\gamma = 1 - \frac{1}{2}\left(\frac{1-\beta}{N-1}\right)^2 \frac{\varepsilon}{M}. \tag{4.37}$$

Proof. By Lemma 4.3.1, we have

$$F(x^k) - \eta^{k+1} \le F(x^k) - \eta^k - \frac{1-\beta}{2(N-1)}\bar{\mu}_k v_k. \tag{4.38}$$

On the other hand,

$$v_k = F(x^k) - \tilde{F}^k(z^k_{j_k}) = F(x^k) - \eta^k + \frac{1}{2}\big\|z^k_{j_k} - x^k\big\|^2_D \ge F(x^k) - \eta^k. \tag{4.39}$$

Combining the last two inequalities, we conclude that

$$\begin{aligned} F(x^k) - \eta^{k+1} &\le F(x^k) - \eta^k - \frac{1-\beta}{2(N-1)}\bar{\mu}_k\big(F(x^k) - \eta^k\big) \\ &= \left(1 - \frac{1-\beta}{2(N-1)}\bar{\mu}_k\right)\big(F(x^k) - \eta^k\big). \end{aligned} \tag{4.40}$$

Consider the definition (4.12) of $\bar{\mu}_k$ in Lemma 4.3.1. If $\bar{\mu}_k = 1$, then $1 - \frac{1-\beta}{2(N-1)}\bar{\mu}_k$ is no greater than the bound (4.37), because $\varepsilon \leq (N-1)M$. Otherwise, $\bar{\mu}_k$ is given by the second case in (4.12). Since the algorithm does not stop, we have $v_k > \varepsilon$, and thus

$$\bar{\mu}_k \geq \frac{(1-\beta)\varepsilon}{(N-1)M}.$$

Substitution to (4.40) yields (4.37). □

Let $x^{(\ell-1)}, x^{(\ell)}, x^{(\ell+1)}$ be three consecutive proximal centers in the algorithm ($\ell \geq 2$). We want to bound the number of iterations with the proximal center $x^{(\ell)}$. To this end, we bound two quantities:

1. The optimal objective value of the *first* subproblem with proximal center $x^{(\ell)}$, whose iteration number we denote by $k(\ell)$:

$$\eta^{k(\ell)} = \min f_{j_{k(\ell)}}(x) + \sum_{i \neq j_{k(\ell)}} \tilde{f}_i^{k(\ell)}(x) + \frac{1}{2}\big\|x - x^{(\ell)}\big\|_D^2. \tag{4.41}$$

We need an upper bound for $F(x^{(\ell)}) - \eta^{k(\ell)}$.

2. The optimal objective value of the *last* subproblem with proximal center $x^{(\ell)}$, occurring at iteration $k'(\ell) = k(\ell+1) - 1$:

$$\eta^{k'(\ell)} = \min f_{j_{k'(\ell)}}(x) + \sum_{i \neq j_{k'(\ell)}} \tilde{f}_i^{k'(\ell)}(x) + \frac{1}{2}\big\|x - x^{(\ell)}\big\|_D^2. \tag{4.42}$$

We need an upper bound for $F(x^{(\ell)}) - \eta^{k'(\ell)}$ which implies the update rule (4.8).

In the following we discuss each issue separately.

Recall that according to the algorithm, $x^{(\ell)}$ is the optimal solution of the last subproblem with proximal center $x^{(\ell-1)}$. Let $f_{j_{k(\ell)-1}}$ be the non-linearized component function of the last subproblem with proximal center $x^{(\ell-1)}$, whose optimal solution is $x^{(\ell)}$. The optimal value of the subproblem (4.5) is

$$\eta^{k(\ell)-1} = f_{j_{k(\ell)-1}}(x^{\ell}) + \sum_{i \neq j_{k(\ell)-1}} \tilde{f}_i^{k(\ell)-1}(x^{(\ell)}) + \frac{1}{2}\big\|x^{(\ell)} - x^{(\ell-1)}\big\|_D^2. \tag{4.43}$$

Lemma 4.4.6. *If a descent step is made at iteration $k(\ell) - 1$, then*

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq \frac{3}{2\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big). \tag{4.44}$$

Proof. The left inequality in (4.28) yields

$$\eta^{k(\ell)} \geq \eta^{k(\ell)-1} - \left\|x^{(\ell)} - x^{(\ell-1)}\right\|_D^2.$$

Since $F(x^{(\ell)}) \leq F(x^{(\ell-1)})$, we obtain

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq F(x^{(\ell-1)}) - \eta^{k(\ell)-1} + \left\|x^{(\ell)} - x^{(\ell-1)}\right\|_D^2.$$

As iteration $k(\ell) - 1$ is a descent step, the update rule (4.8) holds. Thus

$$\begin{aligned} F(x^{(\ell-1)}) - \eta^{k(\ell)-1} &= \Big[F(x^{(\ell-1)}) - \tilde{F}^{k(\ell)-1}(x^{(\ell)})\Big] - \frac{1}{2}\left\|x^{(\ell)} - x^{(\ell-1)}\right\|_D^2 \\ &\leq \frac{1}{\beta}\Big[F(x^{(\ell-1)}) - F(x^{(\ell)})\Big] - \frac{1}{2}\left\|x^{(\ell)} - x^{(\ell-1)}\right\|_D^2. \end{aligned}$$

Combining the last two inequalities we obtain

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq \frac{1}{\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big) + \frac{1}{2}\left\|x^{(\ell)} - x^{(\ell-1)}\right\|_D^2.$$

The right inequality in (4.28) can be now used to substitute $\left\|x^{(\ell)} - x^{(\ell-1)}\right\|_D^2$ on the right hand side to obtain (4.44). □

We can now integrate our results.

Applying Lemma 4.4.3, we obtain the following inequality at *every* null step with prox center $x^{(\ell)}$:

$$\begin{aligned} F(x^{(\ell)}) - \eta^{k} &\geq \frac{2\varphi(\alpha)}{1+2\varphi(\alpha)}\big(F(x^{(\ell)}) - F(x^*)\big) \\ &\geq \frac{2\varphi(\alpha)}{1+2\varphi(\alpha)}\big(F(x^{(\ell)}) - F(x^{(\ell+1)})\big). \end{aligned} \tag{4.45}$$

From Lemma 4.4.6 we know that for $\ell \geq 2$ the initial value of the left hand side (immediately after the previous descent step) is bounded from above by the following expression:

$$F(x^{(\ell)}) - \eta^{k(\ell)} \leq \frac{3}{2\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big). \tag{4.46}$$

Lemma 4.4.5 established a linear rate of decrease of the left hand side. Therefore, the number n_ℓ of null steps with proximal center $x^{(\ell)}$, if it is positive, satisfies the inequality:

$$\frac{3}{2\beta}\big(F(x^{(\ell-1)}) - F(x^{(\ell)})\big)\gamma^{n_\ell - 1} \geq \frac{2\varphi(\alpha)}{1+2\varphi(\alpha)}\big(F(x^{(\ell)}) - F(x^{(\ell+1)})\big).$$

Consequently, for $\ell \geq 2$ we obtain the following upper bound on the number of null steps:

$$n_\ell \leq 1 + \frac{1}{\ln(\gamma)} \ln\left(\frac{4\beta\varphi(\alpha)}{3(1+2\varphi(\alpha))}\frac{F(x^{(\ell)}) - F(x^{(\ell+1)})}{F(x^{(\ell-1)}) - F(x^{(\ell)})}\right). \tag{4.47}$$

If the number n_ℓ of null steps is zero, inequality (4.35) yields

$$\frac{F(x^{(\ell)}) - F(x^{(\ell+1)})}{F(x^{(\ell-1)}) - F(x^{(\ell)})} \leq \frac{F(x^{(\ell)}) - F(x^*)}{F(x^{(\ell-1)}) - F(x^*) - \big(F(x^{(\ell)}) - F(x^*)\big)} \leq \frac{1}{\frac{1}{1-\alpha\beta} - 1}.$$

Elementary calculations prove that both logarithms on the right hand side of (4.47) are negative, and thus inequality (4.47) is satisfied in this case as well.

Suppose there are L proximal centers appearing throughout the algorithm: $x^{(1)}$, $x^{(2)}$, $\ldots$, $x^{(L)}$. They divide the progress of the algorithm into L series of null steps. For the first series, similarly to the analysis above, we use (4.45) and Lemma 4.4.5 to obtain the estimate

$$n_1 \leq 1 + \frac{1}{\ln(\gamma)} \ln\left(\frac{2\varphi(\alpha)}{1+2\varphi(\alpha)} \frac{F(x^{(1)}) - F(x^{(2)})}{F(x^{(1)}) - \eta^1}\right). \tag{4.48}$$

For the last series, we observe that the inequality $F(x^{(\ell)}) - \eta^k \geq \varepsilon/2$ has to hold at each null step at which the stopping test was not satisfied. We use it instead of (4.45) and we obtain

$$n_L \leq 1 + \frac{1}{\ln(\gamma)} \ln\left(\frac{\beta}{3} \frac{\varepsilon}{F(x^{(L-1)}) - F(x^{(L)})}\right). \tag{4.49}$$

We aggregate the total number of null steps for different proximal centers throughout the algorithm and we obtain the following bound:

$$\begin{aligned}\sum_{\ell=1}^{L} n_\ell = \frac{1}{\ln(\gamma)} \Bigg[\ln\left(\frac{2\varphi(\alpha)}{1+2\varphi(\alpha)}\right) + \Big(\ln\left(\frac{\beta}{3}\right) + (L-2)\ln\left(\frac{4\beta\varphi(\alpha)}{3(1+2\varphi(\alpha))}\right) \\ + \ln\left(\frac{\varepsilon}{F(x^{(1)}) - \eta^1}\right)\Bigg] + L\end{aligned} \tag{4.50}$$

Let us recall the definition of γ in (4.37), and denote

$$C = \frac{1}{2}\left(\frac{1-\beta}{N-1}\right)^2 \frac{1}{M},$$

so that $\gamma = 1 - \varepsilon C$. Since $\ln(1-\varepsilon C) < -\varepsilon C$, we derive the following inequality for the number of null steps:

$$\begin{aligned}\sum_{\ell=1}^{L} n_\ell \leq \frac{1}{-\varepsilon C} \Bigg[\ln\left(\frac{2\varphi(\alpha)}{1+2\varphi(\alpha)}\right) + \ln\left(\frac{\beta}{3}\right) + (L-2)\ln\left(\frac{4\beta\varphi(\alpha)}{3(1+2\varphi(\alpha))}\right) \\ + \ln\left(\frac{\varepsilon}{F(x^{(1)}) - \eta^1}\right)\Bigg] + L.\end{aligned} \tag{4.51}$$

Let us now derive an upper bound on the number L of proximal centers. By virtue of (4.7) and (4.8), descent steps are made only if

$$F(x^k) - F(x^*) \geq \beta\varepsilon;$$

otherwise, the method must stop. To explain it more specifically, if $F(x^k) - F(x^*) \leq \beta\varepsilon$, then $F(x^k) - F(z_{j_k}^k) \leq \beta\varepsilon$. If a descent step were made, $F(z_{j_k}^k) \leq F(x^k) - \beta v_k$. Then $\beta v_k \leq \beta\varepsilon$. Since $v_k \leq \varepsilon$, the algorithm would have already stopped, which contradicts our assumption. It follows from Lemma 4.4.4, that

$$(1-\alpha\beta)^{L-1}\big(F(x^{(1)}) - F(x^*)\big) \geq \beta\varepsilon.$$

Therefore,

$$L \leq 1 + \frac{\ln(\beta\varepsilon) - \ln\big(F(x^1) - F(x^*)\big)}{\ln(1-\alpha\beta)}. \tag{4.52}$$

As a result, we have the final bound for the total number of descent and null steps:

$$\begin{aligned} L - 1 + \sum_{\ell=1}^{L} n_\ell \leq \frac{1}{\varepsilon C}\Bigg[& \frac{\ln \frac{F(x^{(1)}) - F(x^*)(1-\alpha\beta)}{\beta\varepsilon}}{\ln(1-\alpha\beta)} \ln\left(\frac{4\beta\varphi(\alpha)}{3(1+2\varphi(\alpha))}\right) \\ & + \ln\left(\frac{1+2\varphi(\alpha)}{2\varphi(\alpha)}\right) + \ln\left(\frac{3}{\beta}\right) + \ln\left(\frac{F(x^{(1)}) - \eta^1}{\varepsilon}\right)\Bigg] \\ & + 2\frac{\ln(\beta\varepsilon) - \ln\big(F(x^{(1)}) - F(x^*)\big)}{\ln(1-\alpha\beta)} + 1. \end{aligned} \tag{4.53}$$

Therefore, in order to achieve precision ε, the number of steps needed is of order

$$L + \sum_{\ell=1}^{L} n_\ell \sim \mathcal{O}\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right).$$

This is almost equivalent to saying that given the number of iterations k, the precision of the solution is approximately $\mathcal{O}(1/k)$.

This chapter is accepted to publish in SIAM Journal on Optimization at the time of defense. The first version was submitted in [DLR15].

Chapter 5

Numerical Illustration

5.1 Application to structured regularized regression problems

In many areas in data mining and machine learning, such as computer vision and compressed sensing, the resulting optimization models consist of a convex loss function and multiple convex regularization functions, called the *composite prior models* in [HZLM11]. For example, in compressed sensing, the linear combination of the total variation (TV) norm and L_1 norm is a popular regularizer in recovering Magnetic Resonance (MR) images. Formally, the models are formulated as follows:

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + \sum_{i=1}^{N} h_i(B_i x), \tag{5.1}$$

where f is the loss function to measure the goodness-of-fit of the data, while the functions h_i are regularization terms. All the functions are convex but not necessarily smooth.

The SLIN algorithm introduced in our paper can be directly applied to solve the general problem (5.1). It can be further specialized to take advantage of additional features of the functions involved. In the following subsection we discuss one such specialization.

5.1.1 Fused lasso regularization problem

The problem is defined as follows:

$$\min_x \frac{1}{2}\big\|b - Ax\big\|_2^2 + \lambda_1\big\|x\big\|_1 + \lambda_2 \sum_{j=1}^{p-1} \big|x_{j+1} - x_j\big|, \tag{5.2}$$

where A is an $m \times n$ matrix, and $\lambda_1, \lambda_2 > 0$ are fixed parameters. This model contains two regularization terms: the *lasso* penalty $h_1(x) = \lambda_1\|x\|_1$, and the *fused lasso* penalty $h_2(x) = \lambda_2 \sum_j |x_{j+1} - x_j|$. We denote the first function as $f(x) := \frac{1}{2}\big\|b - Ax\big\|_2^2$. In models

with a quadratic loss function, we found it convenient to use the matrix $D = \operatorname{diag}(A^T A)$ in the proximal term of the method [LPR14, Sec. 3].

In order to solve each subproblem, we need the gradient of $f(\cdot)$ and subgradients of the regularization functions, which are readily available. Our method requires their explicit calculation at the initial iteration only; at later iterations they are obtained implicitly, as described in Step 1 of the algorithm.

With these, we can solve each subproblem iteratively.

The f-subproblem. Skipping the constants, the f-subproblem has the form:

$$\min_x \frac{1}{2}\|b - Ax\|_2^2 + g_{h_1}^T x + g_{h_2}^T x + \frac{1}{2}\|x - x^k\|_D^2. \tag{5.3}$$

This is a unconstrained quadratic optimization problem and its optimal solution can be obtained by solving the following linear system of equations:

$$(A^T A + D)x = A^T b - g_{h_1} - g_{h_2} + D^T x^k.$$

It can be very efficiently solved by the preconditioned conjugate gradient method with preconditioner D, as discussed in [LPR14, Sec. 3], because the condition index of the system is uniformly bounded. Only matrix–vector multiplications are involved, facilitating the use of a sparse structure of A. After the solution is obtained, the gradient of $f(x)$ and its linearization can be determined by Step 1 of the SLIN algorithm.

The h_1-subproblem. The subproblem is defined as follows (ignoring the constants):

$$\min_x g_f^T x + \lambda_1 \|x\|_1 + g_{h_2}^T x + \frac{1}{2}\|x - x^k\|_D^2. \tag{5.4}$$

This problem is separable in the decision variables, with the following closed-form solution:

$$(x_{h_1})_i = \operatorname{sgn}(\tau_i) \max\left(0, |\tau_i| - \frac{\lambda_1}{d_i}\right), \quad i = 1, \ldots, n.$$

Here $\tau_i = x_i^k - \frac{(g_f)_i + (g_{h_2})_i}{d_i}$.

The solution of the h_1-subproblem gives a new subgradient of h_1 at the minimal point.

The h_2-subproblem. The subproblem is defined as follows (ignoring the constants):

$$\min_x g_f^T x + g_{h_1}^T x + \lambda_2 \sum_{j=1}^{p-1} |x_{j+1} - x_j| + \frac{1}{2}\|x - x^k\|_D^2.$$

Exactly as described in [LPR14], this problem can be equivalently formulated as a constrained optimization problem:

$$\min_{x,z} g_f^T x + g_{h_1}^T x + \lambda_2 \|z\|_1 + \frac{1}{2}\|x - x^k\|_D^2, \text{ subject to } Rx = z, \tag{5.5}$$

with an $(n-1) \times n$ matrix R representing the system $z_j = x_{j+1} - x_j$, $j = 1, \ldots, n-1$. The Lagrangian of problem (5.5) has the form

$$L(x, z, \mu) = g_f^T x + g_{h_1}^T x + \lambda_2 \|z\|_1 + \mu^T (Rx - z) + \frac{1}{2}\|x - x^k\|_D^2,$$

where μ is the dual variable. The minimum of the Lagrangian with respect to z is finite if and only if $\|\mu\|_\infty \leq \lambda_2$. Under this condition, the minimum value of the z-terms is zero and we can eliminate them from the Lagrangian. We arrive to its reduced form,

$$\hat{L}(x, \mu) = g_f^T x + g_{h_1}^T x + \mu^T R x + \frac{1}{2}\|x - x^k\|_D^2.$$

To calculate the dual function, we minimize $\hat{L}(x, \mu)$ with respect to x. After elementary calculations, we obtain the solution

$$\tilde{x}_{h_2} = x^k - D^{-1}(g_f + g_{h_1} + R^T \mu).$$

Substituting it back to the Lagrangian, we obtain the following dual problem:

$$\max_{\mu} -\frac{1}{2}\mu^T R D^{-1} R^T \mu + \mu^T R(x^k - D^{-1} g_f - D^{-1} g_h), \text{ subject to } \|\mu\|_\infty \leq \lambda_2.$$

This problem can be treated as a box-constrained quadratic programming problem, for which many efficient algorithms are available, for example coordinate-wise optimization [LPR14, Sec. 4]. Due to the structure of R, the computational effort per iteration is linear in the problem dimension.

5.1.2 Overlapping group lasso problem

We consider the following problem

$$\min_x \frac{1}{2K\lambda}\|b - Ax\|_2^2 + \sum_{j=1}^{K} d_j \|x_{\mathcal{G}_j}\|_2 \tag{5.6}$$

where $A \in \mathbb{R}^{m \times n}$. This model contains the first function as $f(x) := \frac{1}{2K\lambda}||b - Ax||_2^2$ where parameter $\lambda > 0$ and number of groups K are pre-specified parameters. The second part is

a sum of regularization terms, each penalty function as $h_j(x) = d_j||x_{\mathcal{G}_j}||_2$ where the weights $d_j > 0$ are known parameters. $\mathcal{G}_j \subseteq \{1, \ldots, p\}$ is the index set of a group of variables and $x_{\mathcal{G}_j}$ denotes the subvector of x with coordinates in $\mathcal{G}_j$. This group regularizer has been proven useful in high-dimensional statistics with the capability of selecting meaningful groups of features. The groups could overlap as needed. As the quadratic term has a coefficient of $\frac{1}{2K\lambda}$, the diagonal matrix D in the proximal term of the method is set to $D = \frac{1}{K\lambda}\text{diag}(A^T A)$.

The f-subproblem. The f-subproblem has the form:

$$\min_x \frac{1}{2K\lambda}\|b - Ax\|_2^2 + \sum_{j=1}^{K} g_j^T x + \frac{1}{2}\|x - x^k\|_D^2.$$

It has the same structure as the f-subproblem of the general structured fused lasso example, and can be solved in the same way; just the matrix D is different.

The h_j-subproblem. The h_j-subproblem is defined as follows (ignoring the constants):

$$\min d_j\|x_{\mathcal{G}_j}\|_2 + \langle s, x\rangle + \frac{1}{2}\|x - x^k\|_D^2. \tag{5.7}$$

where $s = g_f + \sum_{j' \neq j} g_{h_{j'}}$; with g_f denoting a subgradient of the function f, and $g_{h_{j'}}$ the subgradients of $h_{j'}$ used in (4.5). To simplify notation, from now on we write $\mathcal{G}$ for $\mathcal{G}_j$.

The decision variables that are outside of the current group $\mathcal{G}$, which we denote $x_{-\mathcal{G}}$, have the following closed-form solution:

$$x_{-\mathcal{G}} = x^k_{-\mathcal{G}} - D^{-1}_{-\mathcal{G}} s_{-\mathcal{G}}.$$

The variables in the current group $\mathcal{G}$ can be calculated as follows. If $x_{\mathcal{G}} \neq 0$, the necessary and sufficient optimality condition for (5.7) is the following equation:

$$\frac{d_j x_{\mathcal{G}}}{\|x_{\mathcal{G}}\|_2} + s_{\mathcal{G}} + D_{\mathcal{G}}(x_{\mathcal{G}} - x^k_{\mathcal{G}}) = 0. \tag{5.8}$$

We denote

$$\frac{d_j}{\|x_{\mathcal{G}}\|_2} = \kappa, \tag{5.9}$$

This leads to

$$x_i = \frac{D_{ii}x_i^k - (s)_i}{\kappa + D_{ii}}, \quad i \in \mathcal{G}. \tag{5.10}$$

Substituting into (5.9), after simple manipulations, we obtain the following equation for κ:

$$\sum_{i \in \mathcal{G}_j} \left(\frac{D_{ii}x_i^k - (s)_i}{1 + \frac{D_{ii}}{\kappa}} \right)^2 = d_j^2. \tag{5.11}$$

Since the left hand side of this equation is an increasing function of κ, we can easily solve it by bisection, if a solution exists. If the columns if A are normalized, then all $D_{ii} = 1$, and equation (5.11) can be solved in closed form.

Letting $\kappa \to \infty$ on the left hand side, we obtain the condition for the existence of a solution of (5.11):

$$\sum_{i \in \mathcal{G}_j} \left(D_{ii} x_i^k - (s)_i \right)^2 > d_j^2. \tag{5.12}$$

If inequality (5.12) is satisfied, κ can be found by bisection and $x_{\mathcal{G}}$ follows from (5.10). If (5.12) is not satisfied, the only possibility is that the optimal solution of (5.7) is $x_{\mathcal{G}} = 0$.

5.2 Numerical Results

In this section, we present some experimental results for problems (5.2) and (5.6). All these studies are performed on an 1.8 GHZ, 4GB RAM computer using MATLAB.

5.2.1 Fused lasso experiments

We evaluate SLIN against six competing methods to assess the effectiveness of our approach. These methods are different in their treatments on four main features, namely, the selection of the block to be optimized, the sufficient improvement test, the choice of the proximal parameter D and the choice of the relaxation parameter. The first one is a direct extension of the alternating linearization method [LPR14], labeled as Cyclical Linearization. It processes the blocks cyclically in a fixed order and performs the improvement test after every block with proximal parameter D, to decide whether to update the current value of x^k. The second method also processes the blocks cyclically in a fixed order but updates x^k after processing all blocks without any test. In the case of two blocks, it would correspond to the Douglas–Rachford operator splitting method of [LM79]; see [LPR14]. We use the name Cyclical Douglas–Rachford (labeled DR_C). The third and fourth methods are similar to DR_C, except that we use over relaxation (with a fixed parameter 1.5) and under relaxation (with a fixed parameter 0.5). We label them as "DR_C over relax" and "DR_C under relax". The fifth modification is Cyclical Douglas–Rachford with proximal parameter ρ, where ρ is set to the average of the diagonal values of $A^T A$. We label it as DR_ρ. Finally, we

compare with [Vu13], a splitting method, which is essentially the same as [Con13]. Table 5.1 summarizes the differences on main features among above methods.

Method Label	Selection Rule	Update Rule	Proximal Parameter D	Relaxation Parameter δ
SLIN	✓	✓	✓	
Cyclical Linearization		✓	✓	
DR_C			✓	
DR_C over relax			✓	✓
DR_C under relax			✓	✓
DR_ρ				

Table 5.1: Main features comparison
over relax: $\delta = 1.5$; under relax: $\delta = 0.5$

We first investigate how these methods approach the optimal objective function value differently. The elements of the matrix design matrix A are generated independently using a Gaussian distribution with mean zero and unit variance. The dependent variables are $b = Ax + \varepsilon$, where ε is a Gaussian noise with variance 10^{-2}. Among the x_j's, 10% equal 1, 20% equal 2, and the rest are zero. For fused lasso, we set the tuning parameters $\lambda_1 = \lambda_2 = 0.5$ and the stopping tolerance $\epsilon = 10^{-3}$, and run all the methods until convergence or reaching maximum iteration count of 1000. The stopping test for all methods but the one of [Vu13] is the same; it is based on test (4.7). For the method of [Vu13], the test is to reach the objective function value at which SLIN stopped, or 1000 iterations, whichever comes first. Then we plot the log(Error) versus the number of iteration in Figure 5.1 ($m = 10000$, $n = 1000$) and Figure 5.2 ($m = 3000$, $n = 4000$). The second case is under-determined and it does not satisfy Assumption 4.4.1.

In the figures, Error is defined as the difference between the optimal value $F(x^*)$ (obtained by SLIN) and the current function value for each method respectively. Since we are plotting log(Error), we delete the last point of SLIN in the figures. In both experiments, SLIN reaches the desired accuracy in a much smaller number of iterations than required by the other methods. Among DR_C, DR_C over relax and DR_C under relax, DR_C under relax is the slowest. Vũ's splitting method makes good improvement at the start, then slows down and cannot achieve the same level of accuracy as SLIN in the maximum allowed number of iterations. When the design matrix is under-determined, Vũ's splitting method

does not converge. DR_ρ is extremely slow or does not converge, for any value of ρ that we tried.

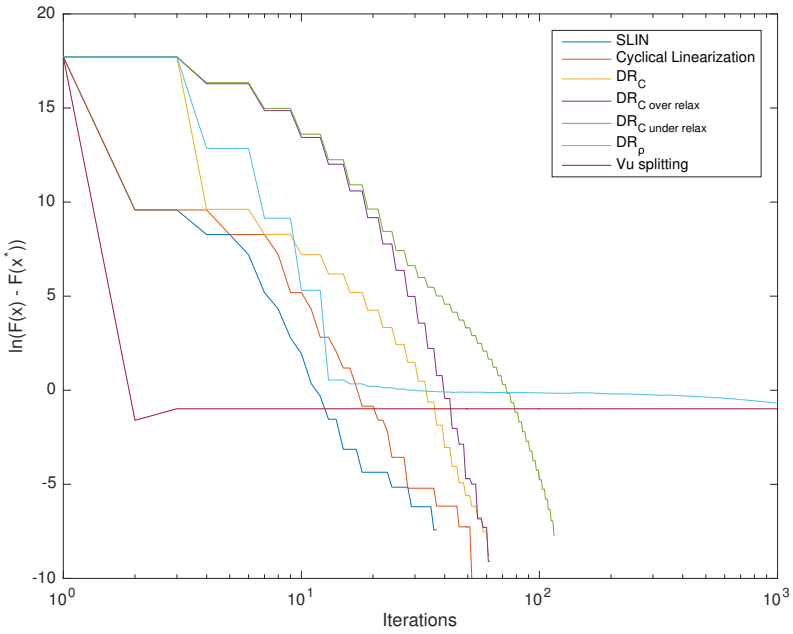

Figure 5.1: Comparison of SLIN and other algorithms on the fused lasso example when $m = 10000, n = 1000$

Next we make running time comparison on these methods. Figure 5.3 and Figure 5.4 report the running time of SLIN and six competing algorithms for problems with different sample sizes (m) and dimensions (n). We run all the algorithms until they satisfy their stopping criteria or use twice the time that SLIN does. Judging from the figures, SLIN performs the best except in a small scale case ($m = 500$, $n = 1000$). Its advantage increases with the scale of the problem.

In Table 5.2, we report the performance of SLIN for different values of the parameter β. We report the average results and their standard deviation. Similar results are observed for experiments with different sample sizes and dimension. Based on these analysis, we use $\beta = 0.5$ in all our experiments.

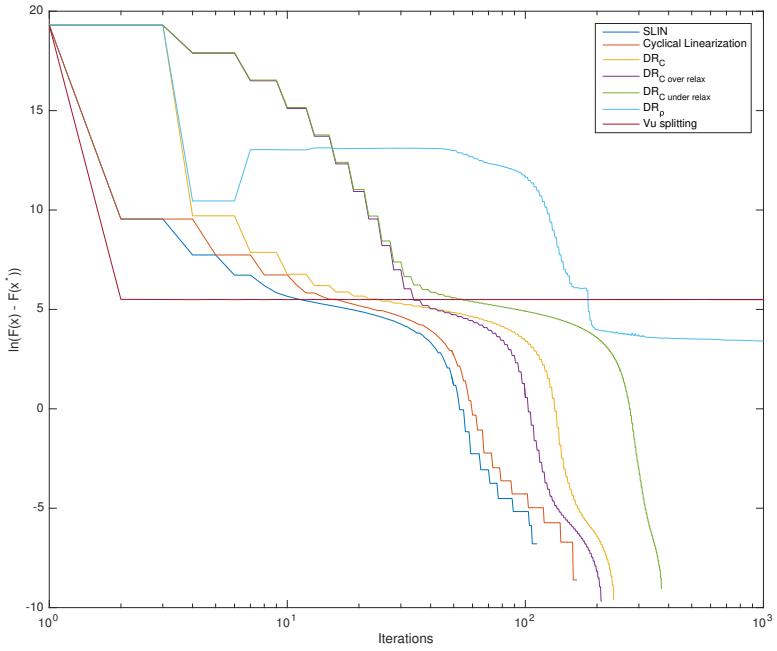

Figure 5.2: Comparison of SLIN and other algorithms on the fused lasso example when $m = 3000, n = 4000$

β	Iterations	Time	Relative Error
0.2	76(25.61)	0.82(0.95)	0.0200(0.0100)
0.5	28(24.70)	0.39(0.64)	0.0080(0.0037)
0.8	74(33.72)	1.08(0.63)	0.0400(0.0200)

Table 5.2: The effect of different values of β in the SLIN algorithm for the fused lasso problem with $m = 1000$ and $n = 300$.

5.2.2 Overlapping group lasso experiments

In this Section we compare SLIN with existing techniques on the tree-structured, fixed order, and random order overlapping group examples. These results demonstrate the flexibility of SLIN and its applicability to problems with complex regularization structures.

Tree-structured overlapping groups

In a tree-structured overlapping group lasso problem, described in [JMOB11b], the groups correspond to the nodes of a tree. The design matrix and input vector are centered and

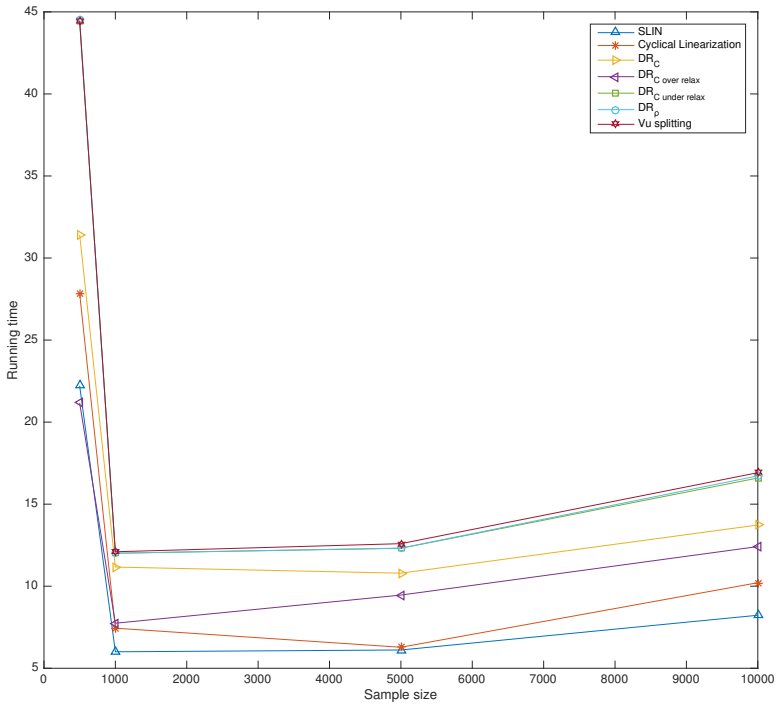

Figure 5.3: Running time of SLIN and other methods on the fused lasso problem as sample size changes when $n = 1000$.

normalized to have unit ℓ_2-norms. We conduct the speed comparisons between our approach and FISTA [JMOB11b]. From Table 5.3 we can see that the SLIN algorithm is faster in terms of both iteration number and computational time.

Parameters	Methods	Iter	Time
$m = 100, n = 10$	SLIN	11.60(0.70)	0.0897(0.0035)
$K = 8$	FISTA	25.20(2.85)	0.1385(0.0138)

Table 5.3: Comparison of SLIN and FISTA on tree-structured overlapping group lasso problem.

Fixed order overlapping groups

We simulate data from a linear model with an overlapping group structure. The entries are sampled from i.i.d. normal distributions, $x_j = (-1)^j \exp(-(j-1)/100)$, and $b = Ax + \varepsilon$, with the noise ε sampled from the standard normal distribution. Assuming that the inputs

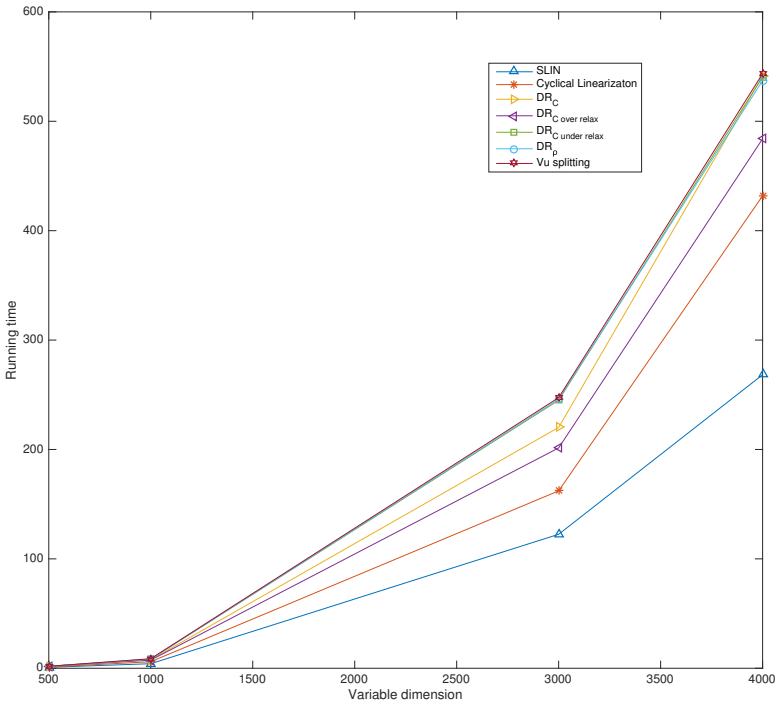

Figure 5.4: Running time of SLIN and other methods on the fused lasso example as dimension changes when $m = 3000$.

are ordered, we define a sequence of K groups of 100 adjacent inputs with an overlap of 10 variables between two successive groups, so that

$$G = \{\{1, \ldots, 100\}, \{91, \ldots, 190\}, \ldots, \{n - 99, \ldots, n\}\}, \tag{5.13}$$

where $n = 90K + 10$. We adopt uniform weights $d_j = 1/K$ and set $\lambda = K/5$.

To demonstrate the efficiency and scalability of the SLIN algorithm, we compare SLIN with several specialized methods for overlapping group lasso problems: PDMM of [CDZ15, WBL14], sADMM or Jacobian ADMM of [DLPY13], PA-APG of [Yu13] and S-APG of [CLK+12]. All experiments are run sequentially, that is, no parallel processing features were exploited. We run the experiments 10 times with different samples of the matrix A; we report the average results. The stopping test for all dual methods is based on the relative change in the iterates: $\|x^{k+1} - x^k\|/\|x^k\| \leq \varepsilon$.

Figure 5.5 plots the convergence of the objective function values *versus* the number of iterations, for the number of groups $K = 100$. For the dual methods PDMM and sADMM,

we report the values of the augmented Lagrangian. They go from super optimal (because the iterates are infeasible) and converge to the optimal value. In Figures 5.6 and 5.7, we vary the group number and sample size and we report the computational time. The SLIN algorithm uses the smallest iteration number but its computational time is worse than that of PDMM, which has been specially coded for problems of this structure. The two "accelerated" methods, PA-APG and S-APG, are the slowest in most tests.

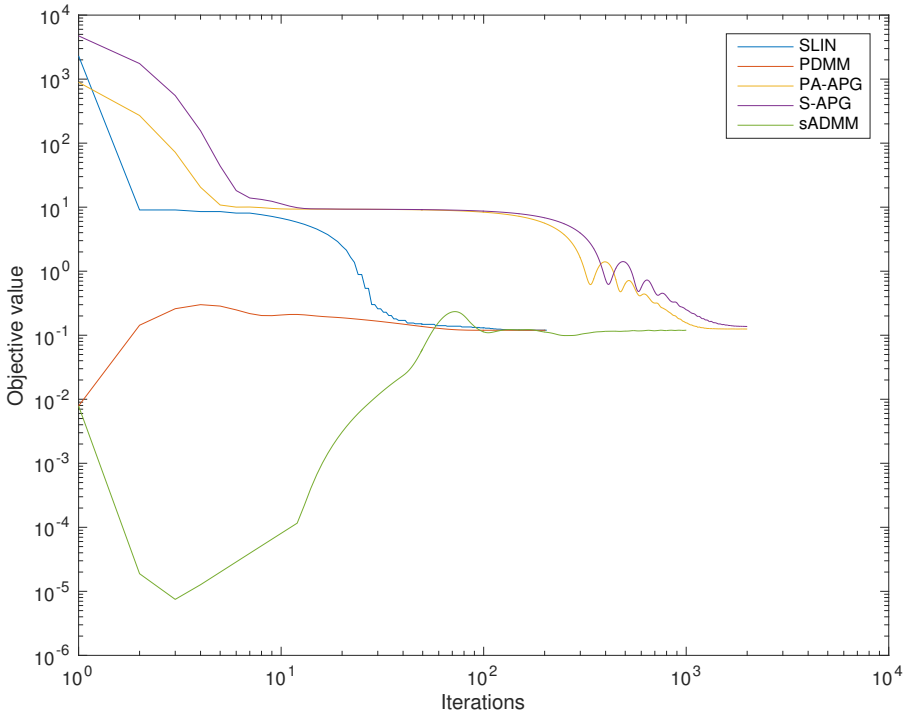

Figure 5.5: Comparison of SLIN and other algorithms on the overlapping group lasso problem when $K = 100, m = 1000$.

Randomly overlapping groups

In the next stage, we conduct additional comparison between SLIN and PDMM on group lasso problems with randomized overlapping, which do not exhibit the regular group structure specified in (5.13).

This type of problem arises in applications such as bioinformatics, where one uses prior information to model potential overlapping of groups of variables. For example, in high throughput gene expression analysis, the number of parameters to be estimated is much

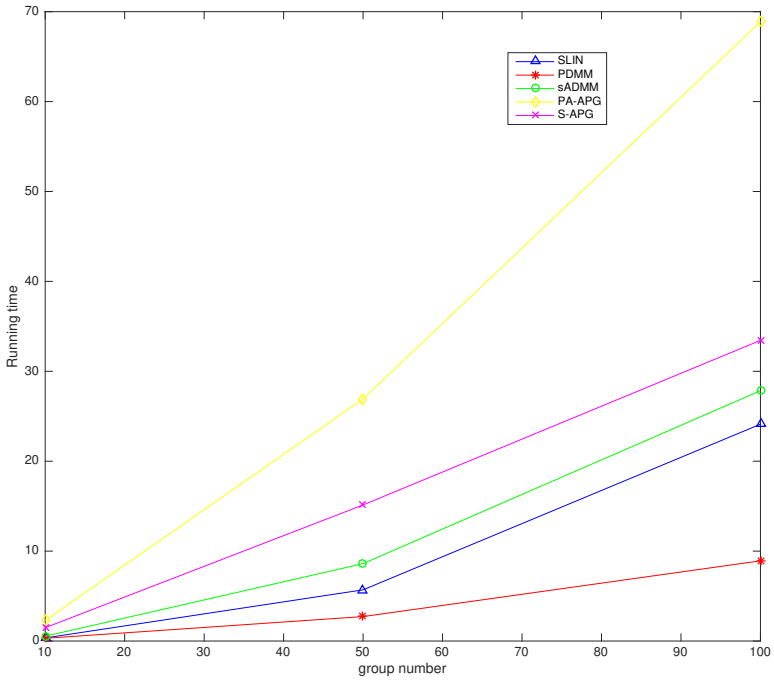

Figure 5.6: Running time of SLIN and other methods on the overlapping group lasso problem as group number changes when $m = 1000$

greater than the sample size. One often utilizes information including gene ontology to define group overlaps among genes, thereby achieving structured regularization [VRMV14]. The resulting overlaps are "arbitrary" (depending on the specific gene ontology) and more complex than the systematic overlapping example described in (5.13). We generate test cases in which the indices in each of the 100 groups were assigned to the n locations. As a result, the number of overlapping variables between the groups was random, and multiple group membership is possible. The performance of the two methods on randomized overlapping group lasso problems is summarized in Tables 5.3 and 5.4.

For fair comparison of the methods, we run PDMM on each instance of the problem. PDMM is set to run to "tol" $= 10^{-4}$ or 2,000 iterations, whichever came first. We set the tuning parameters $d_g = 0.01/K$, and $0.02/K$, respectively. Then SLIN is set to run until the objective function values obtained were as good as that of PDMM. We run the experiments 10 times with different samples of the randomly generated groups; in Tables 5.4 and 5.5 we report the average results and their standard deviations. In all cases, the

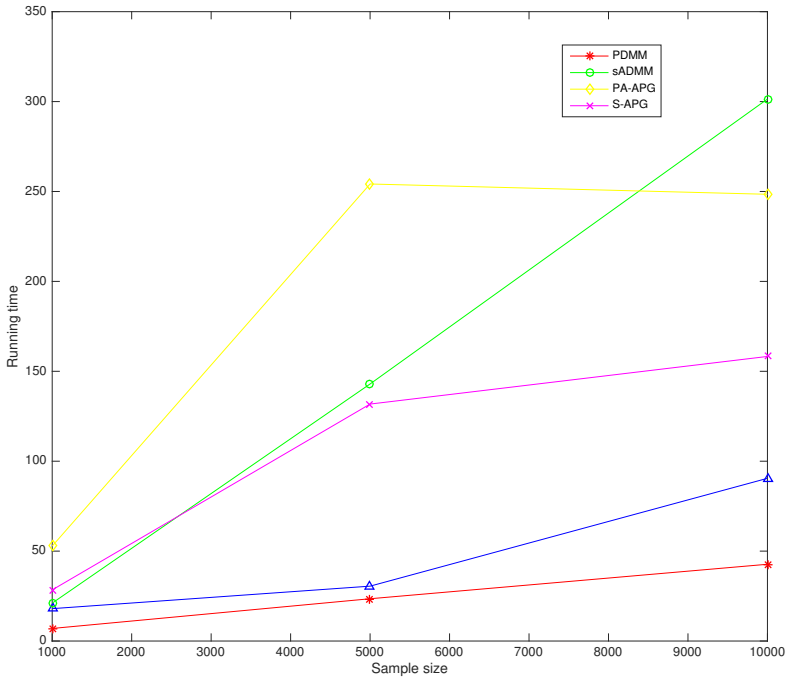

Figure 5.7: Running time of SLIN and other methods on the overlapping group lasso problem as sample size changes when $K = 100$

number of iterations of SLIN is much smaller than that of PDMM. In the determined cases, where $m = 1000$ and $n = 800$, the running time of SLIN is usually better than that of PDMM. In the under-determined cases, where $m = 500$ and $n = 600$, the running time of SLIN is slightly worse than that of PDMM.

In summary, we can conclude that SLIN is a highly efficient and reliable general-purpose method for multi-block optimization of convex nonsmooth functions. It successfully competes with dedicated methods for special classes of problems.

Most of the results presented here were included in a manuscript submitted for publication to a leading optimization journal. A reviewer insisted that we also test the splitting method of section 2.4 which he/she called "the method of Condat." However, the method has not been tested by its authors and no guidance as to its implementation was available. After the results presented here have been included, the associate editor requested to remove the numerical section entirely.

Parameters	Methods	Iter	Time
$m = 1000, n = 800$	SLIN	280 (15.82)	5.42(0.36)
$K = 80, d_g = 0.01/K$	PDMM	638(21.14)	5.48(0.10)
$m = 1000, n = 800$	SLIN	308 (16.11)	5.38(0.43)
$K = 90, d_g = 0.01/K$	PDMM	836(29.33)	7.19(0.39)
$m = 1000, n = 800$	SLIN	331 (13.05)	5.89(0.28)
$K = 100, d_g = 0.01/K$	PDMM	991 (75.51)	8.91(0.81)
$m = 1000, n = 800$	SLIN	306 (11.79)	5.53(0.23)
$K = 80, d_g = 0.02/K$	PDMM	560(59.47)	4.75(0.45)
$m = 1000, n = 800$	SLIN	330 (11.19)	5.63(0.22)
$K = 90, d_g = 0.02/K$	PDMM	620(38.33)	5.65(0.91)
$m = 1000, n = 800$	SLIN	364 (13.77)	6.09(0.31)
$K = 100, d_g = 0.02/K$	PDMM	741(58.06)	4.14 (0.47)

Table 5.4: Comparison SLIN and PDMM in solving the overlapping group lasso of randomly generated groups. Determined cases with $m = 1000$ and $n = 800$.

Parameters	Methods	Iter	Time
$m = 500, n = 600$	SLIN	1280 (39.73)	17.75(0.60)
$K = 80, d_g = 0.01/K$	PDMM	1453(157.07)	12.01(1.03)
$m = 500, n = 600$	SLIN	1119 (57.76)	16.61(1.00)
$K = 90, d_g = 0.01/K$	PDMM	1566(61.63)	13.88(1.63)
$m = 500, n = 600$	SLIN	973 (39.46)	13.72(1.241)
$K = 100, d_g = 0.01/K$	PDMM	1753 (122.27)	16.75(2.64)
$m = 500, n = 600$	SLIN	792 (36.58)	9.79(0.73)
$K = 80, d_g = 0.02/K$	PDMM	968(39.82)	7.53(0.54)
$m = 500, n = 600$	SLIN	722 (23.46)	8.87(0.65)
$K = 90, d_g = 0.02/K$	PDMM	1170(102.16)	9.57(1.00)
$m = 500, n = 600$	SLIN	683 (15.66)	8.01(0.41)
$K = 100, d_g = 0.02/K$	PDMM	1208(70.30)	10.81 (1.21)

Table 5.5: Comparison SLIN and PDMM in solving the overlapping group lasso of randomly generated groups. Underdetermined cases with $m = 500$ and $n = 600$

Chapter 6

Conclusion and Future Research Plan

6.1 Conclusion

We consider the problem of minimizing a sum of several convex non-smooth functions. In this thesis, we introduce a new algorithm called the selective linearization method, which iteratively linearizes all but one of the functions and employs simple proximal steps. The algorithm is a form of multiple operator splitting in which the order of processing partial functions is not fixed, but rather determined in the course of calculations. It proposes one of the first operator-splitting type methods which are globally convergent for an arbitrary number of operators without artificial duplication of variables. This algorithm is a multi-block extension of the alternating linearization (ALIN) method for solving structured non-smooth convex optimization problems.

Global convergence is proved and estimates of the convergence rate are derived. Specifically, under a strongly convex condition, the number of iterations needed to achieve solution accuracy ε is of order $\mathcal{O}\big(\ln(1/\varepsilon)/\varepsilon\big)$. It is a new contribution even in the case of two blocks. The technique invented by us can be also used to derive the rate of convergence of the classical bundle method, for which no convergence rate estimate has been available so far.

We have done extensive comparison experiments in structured regularization problems such as large-scale fused lasso regularization problems and overlapping group lasso problems. The numerical results demonstrate the efficacy and accuracy of the method.

6.2 Future research plan

In the area of convex optimization, many directions of study and improvement for the SLIN algorithm exist. Based on the fact that the subgradient needs to be bounded, SLIN

algorithm can only deal with unconstrained multi-block non-smooth optimization problems. One direction for future study is to design an algorithm for multi-block convex problem with linear operators and analyze the convergence pattern of the algorithm for the following general constrained problems:

$$\min_{x\in\mathbb{X}} F(x) = f_1(x) + \sum_{i=2}^{N} f_i(Mx). \tag{6.1}$$

It is interesting to study the convergence rate of the SLIN algorithm on general convex, but not strongly convex, objective functions.

It is also interesting to study the influence of different linearization order on the component functions on the convergence and convergence rate of the SLIN algorithm. Our current algorithm uses a deterministic rule to specify the order based on the difference the component functions and their linearized approximations. I plan to explore the random linearization order rule and come up with a stochastic version and online versions of SLIN algorithms to facilitate the multi-block objective appearing in many online optimization and stochastic optimization settings.

The study of non-convex optimization algorithms has aroused increasing interest in the field of optimization applied in machine learning and deep learning problems. However, the understanding of convergence of many large-scale non-convex optimization problems is pretty limited. As SLIN is one of the algorithms originally aimed for large-scale convex optimization, I plan to study the convergence performance of (variants of) the SLIN algorithm on some classes of non-convex optimization problems.

On the other hand, the simplicity and efficiency of the selective linearization framework suggests that it can be extended to solve many problems that are currently of interest for many other machine learning and data mining problems including low rank matrix completion and compressed MR image reconstruction, and even some deep learning problems.

References

[BC11] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.

[Ben09] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2(1):1127, 2009.

[Ber03] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientif, with A. Nedíc and A. E. Ozdaglar, 2003.

[Ber15] D. P. Bertsekas. Incremental aggregated proximal and augmented lagrangian algorithms. Technical Report Report LIDS-3176, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass., 2015.

[BGLS03] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. Sagastizábal. Numerical optimization. theoretical and practical aspects. 2003.

[BPC+10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[CDZ15] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos. An augmented lagrangian method for distributed optimization. *Mathematical Programming*, 152(1-2):405–434, 2015.

[CE16] P. Combettes and J. Eckstein. Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions. *Mathematical Programming*, pages 1–28, 2016.

[CHYY14] C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, pages 1–23, 2014.

[CLK+12] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics*, 6(2):719–752, 2012.

[Com09] P. L. Combettes. Iterative construction of the resolvent of a sum of maximal monotone operators. *J. Convex Anal.*, 16(3-4):727–748, 2009.

[Con13] L. Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

[CP11] P. L. Combettes and J. C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pages 185–212. Springer, 2011.

[CP12] P. L. Combettes and J. C. Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators. *Set-Valued and Variational Anal.*, 20:307–330, 2012.

[DHHH13] C. Demiralp, E. Hayden, J. Hammerbacher, and J. Heer. Exploring high-dimensional rna sequences from in vitro selection. *IEEE Biological Data Visualization*, 2013.

[DLPY13] W. Deng, M. J. Lai, Z. Peng, and W. Yin. Parallel multi-block admm with o (1/k) convergence. Technical report, 2013.

[DLR15] Y. Du, X. Lin, and A. Ruszczynski. Selective Linearization For Multi-Block Convex Optimization. *ArXiv e-prints*, November 2015.

[DR56] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.

[DR17] Y. Du and A. Ruszczyński. Rate of convergence of the bundle method. *J Optim Theory Appl*, 2017.

[EB92] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55(3, Ser. A):293–318, 1992.

[Eck17] J. Eckstein. A simplified form of block-iterative operator splitting and an asynchronous algorithm resembling the multi-block alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, pages 1–28, 2017.

[ES09] J. Eckstein and B. F. Svaiter. General projective splitting methods for sums of maximal monotone operators. *SIAM J. Control Optim.*, 48(2):787811, 2009.

[EY15] J. Eckstein and W. Yao. Understanding the convergence of the alter- nating direction method of multipliers: Theoretical and computational perspectives. *Pacific Journal of Optimization*, 11(4):619644, 2015.

[GM75] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975.

[GM76] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[GMS13] D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, (141):349–382, 2013.

[GO09] T. Goldstein and S. Osher. The split Bregman method for $L1$-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.

[GT89] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, volume 9. SIAM, 1989.

[HL17] M. Hong and Z. Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1):165199, 2017.

[HUL93] 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.

[HY15] B. He and X. Yuan. On the convergence rate of douglas-rachford operator splitting method. *Mathematical Programming*, 153(2):715722, 2015.

[HZLM11] J. Huang, S. Zhang, H. Li, and D. Metaxas. Composite splitting algorithms for convex optimization. *Computer Vision and Image Understanding*, 115(12):1610–1622, 2011.

[JMOB11a] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12(Jul):2297–2334, 2011.

[JMOB11b] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12(1):2297–2334, 2011.

[Kiw83] K. C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. *Mathematical Programming*, 27(3):320–341, 1983.

[Kiw85] K. C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*, volume 1133 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1985.

[Kiw95] K. C. Kiwiel. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Mathematical Programming*, 69(1-3):89–109, 1995.

[KRR99] K. C. Kiwiel, C. H. Rosa, and A. Ruszczyński. Proximal decomposition via alternating linearization. *SIAM Journal on Optimization*, 9(3):668–689, 1999.

[Kwa15] T. Kwartler. Intro to text mining using tm, opennlp and topic models. *Open Data Science Conference*, 2015.

[Lan15] G. Lan. Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization. *Mathematical Programming*, 149(1-2):1–45, 2015.

[Lem78] C. Lemaréchal. Nonsmooth optimization and descent methods. *Research Report 78-4, International Institute of Applied Systems Analysis, Laxenburg, Austria*, 1978.

[LM79] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

[LMZ15] T. Lin, S. Ma, and S. Zhang. On the global linear convergence of the admm with multiblock variables. *SIAM Journal on Optimization*, 25(3):14781497, 2015.

[LNN95] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69(1-3):111–147, 1995.

[LPR14] X. Lin, M. Pham, and A. Ruszczyński. Alternating linearization for structured regularization problems. *The Journal of Machine Learning Research*, 15(1):3447–3481, 2014.

[Mif82] R. Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. *In D. C. Sorensen and R. J. B. Wets, editors, Nondifferential and Variational Techniques in Optimization*, 17:77–90, 1982.

[PR55] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.

[Roc70] A. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Roc76] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[Rus86] A. Ruszczyński. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35(3):309333, 1986.

[Rus06] A. Ruszczyński. *Nonlinear optimization*, volume 13. Princeton University Press, 2006.

[RW91] R. T. Rockafellar and R. J. B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147, 1991.

[TW07] R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics*, pages 1–12, 2007.

[VRMV14] S. Villa, L. Rosasco, S. Mosci, and A. Verri. Proximal methods for the latent group lasso penalty. *Computational Optimization and Applications*, 58(2):381–401, 2014.

[Vu13] B. C. Vu. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3):667–681, 2013.

[WBL14] H. Wang, A. Banerjee, and Z. Q. Luo. Parallel direction method of multipliers. pages 13–64. Neural Information Processing System (NIPS), 2014.

[Yu13] Y. L. Yu. Better approximation and faster algorithm using the proximal average. In *Advances in Neural Information Processing Systems*, pages 458–466, 2013.

[YX11] G. B. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Comput. Statist. Data Anal.*, 55(4):1552–1569, 2011.

[ZGWY15] J. Zhou, P. Gong, Z. Wang, and J. Ye. Mining structured sparsity beyond convexity. *ICDM Tutorial*, 2015.