University of California
Santa Barbara

Topics in Optimization: Restarted Moment Based Methods, Underdetermined Systems and Trade Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Can Kizilkale

Committee in charge:

Professor Shivkumar Chandrasekaran, Chair
Professor Omer Egecioglu
Professor John Gilbert

September 2019

The Dissertation of Can Kizilkale is approved.

Professor Omer Egecioglu

Professor John Gilbert

Professor Shivkumar Chandrasekaran, Committee Chair

August 2019

Topics in Optimization: Restarted Moment Based Methods, Underdetermined Systems and Trade Networks

by

Can Kizilkale

To my Parents.

Acknowledgements

I'd like to thank Professor Egecioglu for all the help and guidance during my stay here, Professor Chandrasekaran, a kind hearted, intelligent mathematician whom I benefited from the discussions we had during my research and Professor Gilbert for being a committee member.

I want to thank Professor Mustafa Pinar for his guidance and support in my troubling years, it was always a great experience sharing ideas, making research which led to several publications. He is not only a great researcher he is also a very good human being and I am very lucky to know him. I wish we started making research during my undergraduate years.

I am grateful to Professor Rakesh Vohra whom I had the privilege to make research with. Not only he went out of his way to host me at University of Pennsylvania, his vision, generosity, ideas inspired me. I wish I did the right decision back then but I am hoping that we are going to continue to work on many more interesting problems together.

Curriculum Vitæ
Can Kizilkale

Education

Ph.D. in Computer Science , University of California, Santa Barbara.
M.S. ECE, Koc University.
B.S. EE, Bilkent University

Publications

Work in Progress

[1] Rakesh Vohra, Can Kizilkale, *Trade Networks under complex constraints.*

Work under review

[1] Can Kizilkale, Shivkumar Chandrasekaran, Mustafa C. Pinar and Ming Gu, *Gradient Based Adaptive Restart is Linearly Convergent.*
[2] Can Kizilkale,Mustafa C. Pinar, *Sparsest Solution to an Underdetermined System of Linear Equations via Penalized Huber Loss.*

Journal Publications

[1] Mustafa C. Pinar and Can Kizilkale, *Robust Screening Under Distribution Ambiguity*, (Mathematical Programming Ser. A, vol. 163(1), 273-299, April 2017.).
[2] Can Kizilkale, Omer Egecioglu and C. K. Koc, *A Matrix Decomposition Method for Optimal Normal Basis Multiplication*, (IEEE Transactions on Computers, Volume 65 Issue 11, November 2016 Page 3239-3250).
[3] Alper T. Erdogan and Can Kizilkale, *Fast and Low Complexity Blind Equalization via Subgradient Projections'*, in IEEE Transactions on Signal Processing, Vol: 53, pp: 2513-2524, July 2005.

Conference Publications

[1] Can Kizilkale and Alper T. Erdogan, *A Fast Blind Equalization Method Based On Subgradient Projections'*, in Proceedings of the IEEE ICASSP, Vol: 4, pp:873-876, May 2004.
[2] Can Kizilkale and Alper T. Erdogan, *A Moving Window Approach for Blind Equalization Using Subgradient Projections*, in Proceedings of IEEE SIU, pp: 196-199, April 2004.

[3] Can Kizilkale, *Cost Sharing of Multiple Non-Excludable Public Goods*, presented at 9th conference on Web and Internet Economics held at Harvard University,ACM 1 transactions on Economics and Computation, 2013.

Technical Reports / Other

[1] Can Kizilkale, *A Fair, Sequential Multiple Objective Optimization Algorithm*, eprint in Optimization Online, August 2012 (Technical Report)
[2] Can Kizilkale, *A Family of Polytopes in the 0/1-cube having Gomory-Chvatal rank at least ((1+1/6)n-4)*, e-print in Optimization Online, July 2012.
[3] Can Kizilkale, *Fast Blind Equalization Using Subgradient Based Algorithms*, M.Sc. Thesis, Koc University, May 2005
[4] Can Kizilkale, *Eliminating Duality Gap by Sigma Penalty*, May 2010 (e print in Optimization online July 2012)

Research Interests: Optimization, Game Theory, Matrix Analysis.

Abstract

Topics in Optimization: Restarted Moment Based Methods, Underdetermined Systems and Trade Networks

by

Can Kizilkale

This thesis consists of three parts focusing on three different problems. All of these problems have optimization at heart yet they are related to different branches of optimization.

Our first problem is on the convergence rate of two moment based methods, Polyak's Heavy Ball and Nesterov's Accelerated Gradient, employing adaptive restart. We introduce two criteria and show that under such criteria these methods have linear convergence. Strongly convex functions satisfy these criteria hence the results are valid for such functions. To the best of our knowledge, our result is the first convergence result for the adaptive restart scheme. We then introduce a novel restarting criteria which are highly tunable and also satisfies linear convergence.

Our second problem is on computation of the sparsest solution to an underdetermined system of linear equations. We introduce an extrapolation procedure which computes the sparsest solution from a penalized relaxation of the problem via Huber function. This extrapolation procedure uses a condition called sign constancy and we show that if one works with extreme points this can be removed. We present necessary and sufficient conditions for the recovery of the sparsest solution by penalized Huber loss function and ties among different solutions.

Our last problem of concern is on Network flows. In this work in progress, we investigate existence and computation of the equilibrium in trade networks with constraints.

To the best of our knowledge, these networks are only investigated under simple capacity constraints on each link. Here we first start with a negative result where we give a counterexample where the link capacities live in a polymatroid yet there is no integer equilibrium point in the feasible region. We then move on to investigate the cases where the constraints are less restrictive.

Contents

Chapter 1

Convergence rate of Restarted Moment based Algorithms

1.1 Introduction

Speeding up gradient based algorithms via momentum is a well-known technique. Polyak's heavy-ball method [1] and Nesterov's accelerated gradient algorithm [2] are probably the two most famous algorithms using momentum.

Nesterov's method is well-known for achieving fast convergence despite not being more complex than the classical gradient descent algorithm. Although the algorithm was introduced more than three decades ago, it became very popular in the last decade due to its benefits in solving large scale problems in sparse signal recovery, machine learning, composite function optimization, etc., where higher order methods become infeasible.

The idea behind accelerated gradient scheme is building up momentum to increase the convergence rate. At each step instead of just taking into account the gradient we also take into account the momentum vector which is essentially a weighted sum of all the previous steps. The momentum vector contains some second order information about

the objective function which leads to accelerated convergence when used correctly.

An important problem with the accelerated gradient algorithm (and momentum based methods in general) is that it exhibits non-monotonic convergence behavior. This behavior seems to be periodic and lowers the convergence rate. An intuitive explanation of this behavior is that, as the momentum increases, the algorithm takes much larger steps towards the optimum point, leading to faster decrease in the function value, until the point where the momentum vector makes the iterates move away from the optimum(overshoots) causing the function value to increase until the gradient of the objective function nullifies and corrects the direction.

One noteworthy observation is that when step sizes are chosen small enough the algorithm exhibits monotonic convergence until the first point of overshoot. The original algorithm lets the gradient slow down the algorithm once it overshoots. Yet we can obviously do better if we slow it down or stop it "artificially" when overshoot happens. Instead of slowing the algorithm using the gradient we restart it, which erases the history and starts the algorithm afresh using the current iterate as the initial point. If we know the condition number then one can exploit the periodic behavior of the non-monotonicity and employ periodic restarts at those points to achieve linear convergence [3]. When we do not have that information though, it is difficult to decide on the right periodicity.

Some of the tests for detecting overshoot are the exact non-monotonicity test [4], and the gradient-mapping test [5], both of which seem to work well in practice.

1.1.1 Contributions

We shall first focus on the gradient-mapping test based restart and prove that it exhibits linear convergence under conditions which are more relaxed than strong convexity. To the best of our knowledge our paper is the first contribution to establish a conver-

gence rate result for gradient-based method with restarts. Prior analysis was restricted to quadratic functions only. More recent analyses of accelerated gradient methods have been based on ODEs [3, 6]. The main idea is to analyze the continuous case, where step size is arbitrarily small, and then expand the analysis by quantizing the continuous path. However, we use here the classical approach in proving the convergence rate. We show that with the gradient based restart condition the algorithm becomes monotonic. The momentum vector in the worst case grows like $O(\sqrt{k})$ (c.f. Corollary 1.6.2), and even in this case we have shown linear convergence rate (for additional experimental results on the effectiveness of this restart rule the reader can also refer to [4, 5]).

Our results are obtained under conditions more general than strong convexity. We introduce three conditions which are sufficient for linear convergence. There has been quite an extensive research on achieving linear convergence without strong convexity (such as [7, 8, 9]). The main ingredient used to achieving linear convergence is quadratic growth condition which is a quadratic lower bound on the objective $f(x)$ with respect to the shortest distance from x to the optimal set. Our condition is different from quadratic growth condition. Instead of assuming a bound on the function we check what happens in between subsequent iterates. We then show that these conditions can be satisfied without the need of strong convexity.

Finally we will introduce a new restarting condition, *cone based restart* which also achieves linear convergence under our conditions. This scheme gives similar performance to adaptive gradient scheme yet it uses the gradient in the beginning of each restart as pivot which shows that the information on the direction of acceleration is available to us in the beginning of each restart.

1.2 Preliminaries

Convex optimization is a deep and well studied subject. Although one can try to overview the fundamental concepts we believe there are many well written resources available and it is best to point the reader to one of them instead of giving a shallow summary. In our opinion Nemirovski's lecture notes ([10]) is a fantastic read and a very comprehensive resource.

Throughout this chapter we are interested in solving the general unconstrained convex optimization problem,

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

where $f : R^n \to R$ is a convex function with L-Lipschitz continuous gradient where gradient being L-Lipschitz continuous is defined as follows (for the rest of the paper we will be denoting euclidean norm with $\|.\|$ unless otherwise is stated).

Definition 1 *The gradient of f is L-Lipschitz continuous if there exists a constant $L > 0$ such that $\forall\ x, y \in dom(f)$*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|. \tag{1.1}$$

L-Lipshitz continuity is an essential and reasonable assumption. Without it(L-Lipshitz continuity) the change in the gradient at each step may be unbounded and it becomes almost impossible to come up with a convergence guarantee (for most of the functions we are interested in difference of gradients in between two points will be bounded by a function of the distance between the two).

While not as reasonable as the L-Lipshitz continuity, Strong convexity is another very important property. There are several equivalent definitions of strong convexity. We will use the following one.

Definition 2 *A function $f : R^n \to R$ is strongly convex if* $\forall\ x, y \in dom(f)$

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + (\mu/2)\|y - x\|^2, \tag{1.2}$$

for some constant $\mu > 0$.

This property essentially tells us that the function is lower bounded by a quadratic function with strictly positive eigenvalues.

1.2.1 Gradient Descent

The most fundamental but still capable optimization method is gradient descent. The idea is at each step to move in the opposing direction of the gradient direction by some selection of stepsize to reach a better solution.

Algorithm 1: Gradient Descent

Choose $x_{-1} \in R^n$
$x_0 = x_{-1}$
for $k \geq 0$ **do**
  Pick α_k
  $x_{k+1} = x_k - \alpha_k \nabla f(y_k)$
end for

For functions with continuous gradients α_k can be taken as a small enough constant. For arbitrary convex functions as long as $\lim \alpha_k = 0$ and $\sum_k \alpha_k = \infty$ convergence is guaranteed. The convergence rate of gradient descent for arbitrary convex functions is known to be $O(k^{-1})$.

1.2.2 Accelerated Gradient

Nesterov's accelerated gradient algorithm is an instance of the general momentum based algorithms. This algorithm produces a sequence of iterates $x_k, y_k \in R^n$ by the

following update rule.

Definition 3 *Generalized accelerated gradient update rule:*

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \tag{1.3}$$

$$x_{k+1} = y_k - \alpha_k \nabla f(y_k), \tag{1.4}$$

where the term $\beta_k(x_k - x_{k-1})$ *is the* momentum *term at each step.*

The optimum selection for β_k is by solving another equation at each step. The original algorithm is given in (2).

Algorithm 2: Accelerated Gradient Algorithm

Choose $x_{-1} \in R^n$
$x_0 = x_{-1}$, $y_0 = x_0$, $\theta_0 = 1$, $q = [0, 1]$
for $k \geq 0$ **do**
  $x_{k+1} = y_k - t_k \nabla f(y_k)$
  θ_{k+1} solves $\theta_{k+1}^2 = (1 - \theta_{k+1})\theta_k^2 + q\theta_{k+1}$
  $\beta_{k+1} = \theta_k(1 - \theta_k)/(\theta_k^2 + \theta_{k+1})$
  $y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k)$
end for

Nesterov has shown ([2]) that accelerated gradient (we will refer to Nesterov's algorithm as "accelerated gradient" for the rest of the paper) has a guaranteed convergence rate of $O(k^{-2})$ for general convex functions under L-Lipshitz continuity. However, for strongly convex functions, if the condition number μ and Lipschitz constant L are known, it can be improved to linear convergence, $O(c^{-k})$ (c is some constant that is greater than 1) [11]. Unfortunately both are unknown in many problems. Moreover, it is frequently impractical to estimate μ.

1.2.3 Heavy-Ball Method

Although Nesterov's algorithm is the most popular one it should be noted that the algorithm that made algorithms using momentum (from now on we will refer to the difference between two back to back iterates, $x_{k+1}-x_k$, as the **momentum** at step $k+1$) a viable option for optimization is probably Polyak's Heavy-ball method([1]) which has the following update rule.

Definition 4 *Heavy-ball update rule:*

$$x_{k+1} = x_k + \beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k).$$

The complete algorithm is as follows.

Algorithm 3: Heavy-Ball Method

Choose $x_{-1} \in R^n$
$x_0 = x_{-1}$, $y_0 = x_0$, $\theta_0 = 1$
for $k \geq 0$ **do**
  $x_{k+1} = x_k + \beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k)$
end for

1.3 Why Momentum Based Methods?

We start by discussing what makes Momentum based approaches so popular. Although gradient descent is versatile and simple to implement it can be painfully slow at times. It is especially prone to what we call "zigzagging"(figure 1.1).

Using momentum allows us to tackle this problem very well, it "corrects" the direction of descent by smoothing out consequent gradients by the momentum term. It also increases the stepsize. Both of these contribute to the convergence increase we observe in figure 1.2.

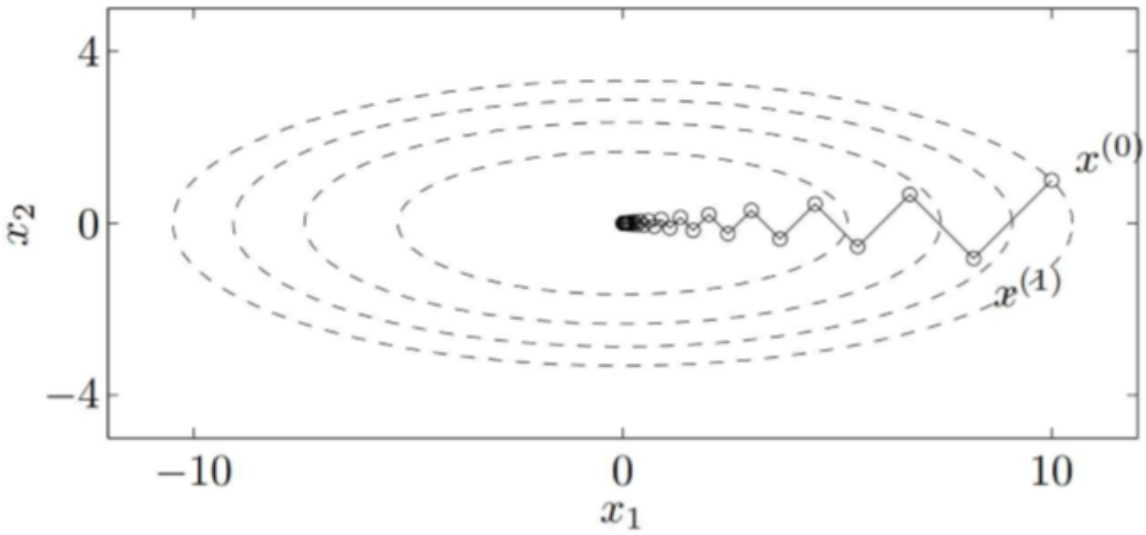


Figure 1.1: Zigzagging effect on Gradient Descent

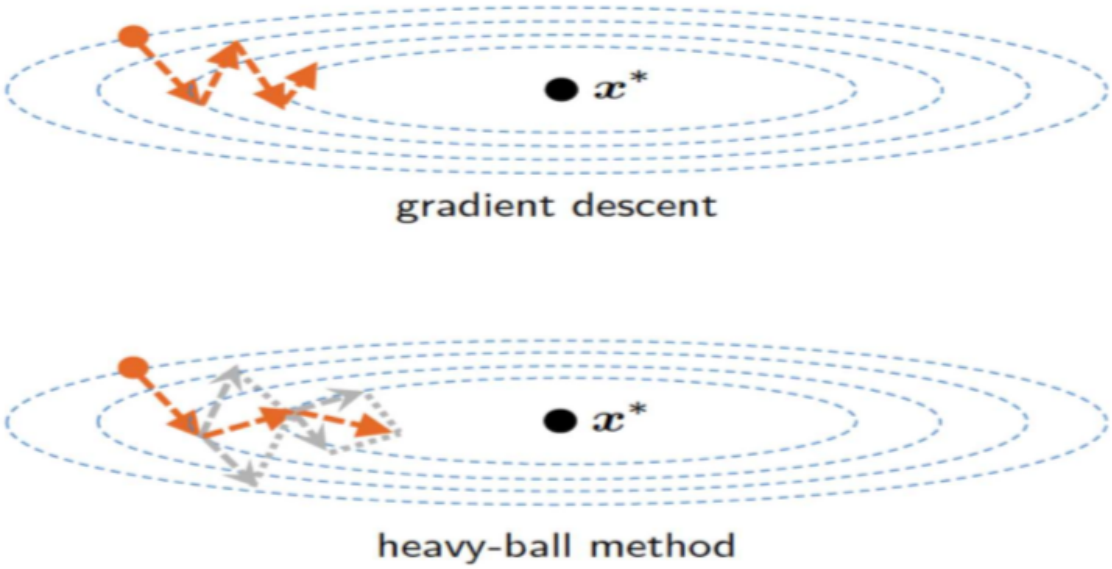
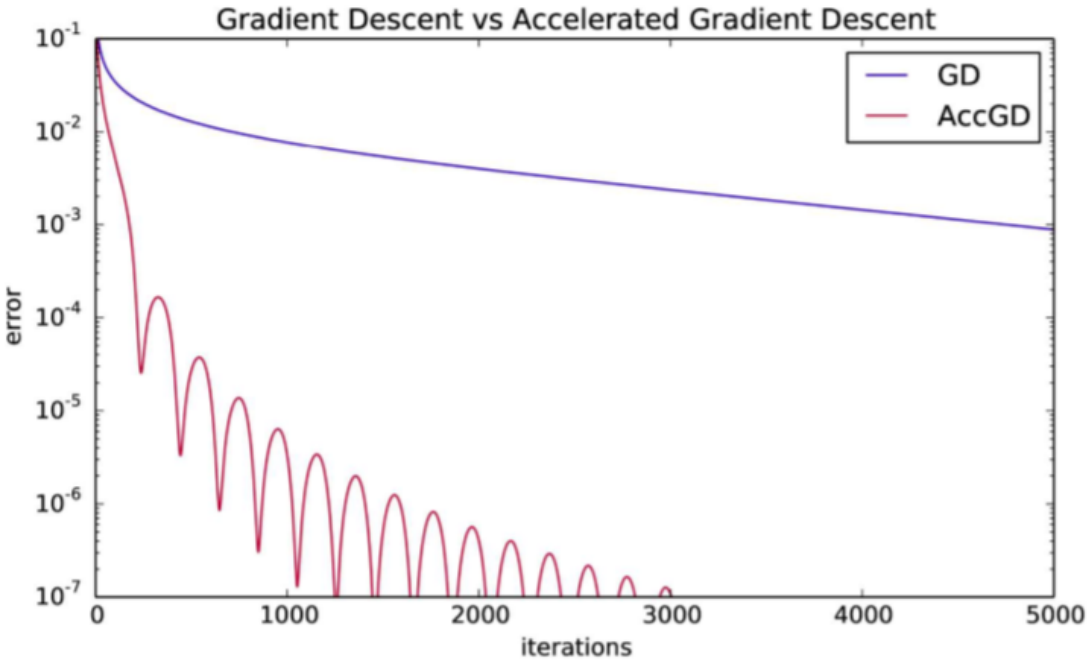


Figure 1.2: Direction correction

Figure 1.3: Accelerated Gradient vs Gradient Descent



Source: http://blog.mrtz.org/2014/08/18/robustness-versus-acceleration.html

Accelerated gradient is not a monotonically decreasing method. By looking at figure (1.3) we can make two interesting observations. The first one is that Accelerated gradient is asymptotically faster than gradient descent and the second one is the non-monotonic behavior seems to be periodic.

The most intuitive explanation for such non-monotonic behavior is that while the stepsize increases, because of momentum build-up, it causes faster convergence until the point where momentum vector points away from the optimum. For convex functions once this happens moving in the direction of the momentum will simply increase the function value. In the next section we will see a strong method to deal with this behavior.

1.4 General Restarted Momentum Based Gradient Descent

The somewhat periodic behavior we observed in figure (1.3) can be dealt in different ways. If the condition number is known then one can select the step size so that there is no overshooting and the algorithm converges monotonically. This is most of the time not feasible since such information is not available. Another way of dealing with this behavior is to zero the momentum when we detect overshooting. We will call this restarting the algorithm. The algorithm is given as 3.

Algorithm 4: Generalized Restarted Momentum Method

Choose $x_{-1} \in R^n$
$x_0 = x_{-1}$
for $k \geq 0$ **do**
  $z_{k+1} = \beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k)$
  if restart condition is satisfied **then**
    $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
  else
    $x_{k+1} = x_k + z_{k+1}$
  end if
end for

1.4.1 Several Restarting Criteria

There are several different ways for restarting the algorithm. We are going to summarize the major ones as follows.

- *Periodic restart:* Restarts the algorithm by a predefined restarting interval. Without the knowledge of condition number not possible to find the right period
- *Exact monotonicity test:* Checks if there is a decrease in the objective value, if not restarts. Prone to numerical errors when the iterate is too close the optimum.

- *Gradient-mapping restart:* Checks the inner product of momentum vector and the current gradient. We will analyze this one in the next section
- *Cone Based restart:* Checks if the current gradient falls in a cone centered around a pivot. (Our proposed method).

1.5 Gradient-Mapping restart condition

The restart condition that we will anaylze first is the test called "gradient-mapping restart" which was proposed in [5]. The idea is to detect whether the ascent direction which is determined by the momentum vector has a positive projection on the gradient(5).

Definition 5 *Gradient-mapping restart condition:*

$$\nabla f(x_k)^T(x_k - x_{k-1}) > 0.$$

The algorithm that we are going to plug this in will be the Heavy-Ball based one (algorithm 4). The reason for this is making the analysis cleaner. Afterwards we will explain how to extend the analysis to the accelerated gradient case.

Before going on to the analysis part where we show that gradient based restart has linear convergence and the corresponding criteria for this result we should give two graphs from [5] to motivate the intuition. In figure 1.4 we can see that when for the regular accelerated gradient there is periodic overshooting where for the restarted case(the green line) once the gradient-mapping condition is satisfied (and it detects overshooting quite successfully) the algorithm is restarted and changes the direction to point towards the optima. The purple path shows the optimal path when we have the condition number plugged in to get maximum speed up from accelerated gradient.

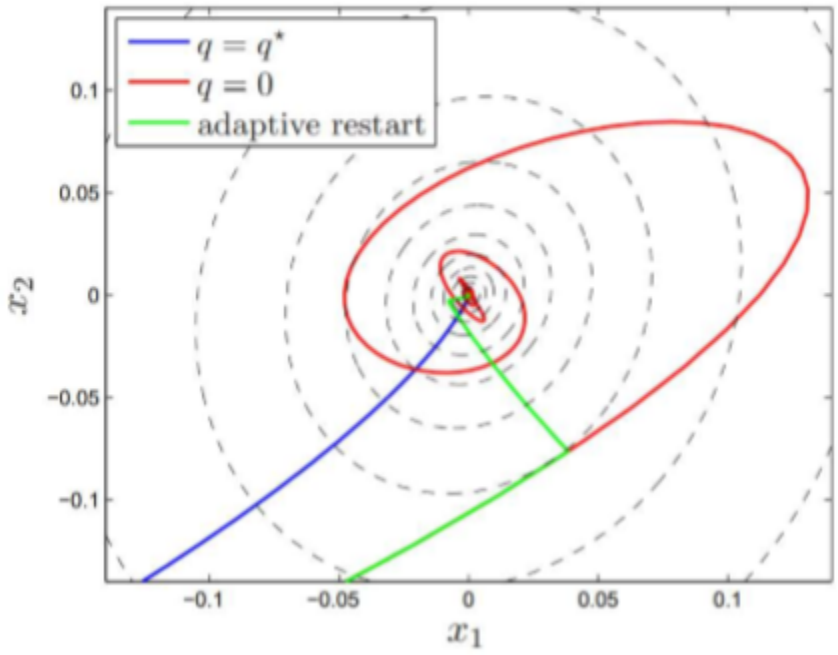


Figure 1.4: When Overshoot happens ([5])

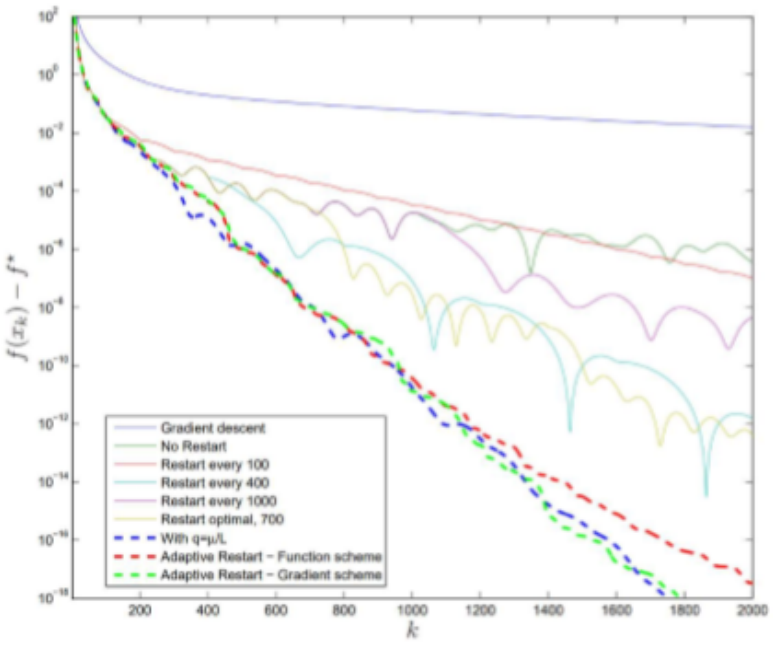


Figure 1.5: Comparison of different restarting schemes ([5])

In figure(1.5) we can see a comparison of different restarting criteria. For the periodic restarts we can see that the performance gain is not much and it is not possible to fix the non-monotonic behavior without additional information. The gradient based and function based restart schemes follow the optimum path (the one where we use the condition number to tune the step size) up to a neighborhood of the optimum point afterwards the functional restart begins to degrade because numerical issues.

1.6 Convergence rate of Momentum based Gradient Descent with Gradient-Mapping restart

In this section we will lay down our major contribution for this chapter. To the best of our knowledge this is the first proof that shows that gradient-mapping restart has linear convergence under strong convexity. We also introduce two new criteria where our

result holds and show that these criteria is satisfied by strong convexity but are weaker.

We are going to denote the optima, in this case a minimizer, as x^*. We will assume that the objective function $f(x)$ is convex, smooth with L-Lipshitz gradient.

1.6.1 Criteria for Linear convergence

We are going to introduce the following criteria which going to be used for proving convergence.

$$\|\nabla f(x)\|^2 \geq (f(x) - f(x^*))/M, \tag{1.5}$$

for some $M > 0$,

$$f(x_k) - f(x_{k+1}) \geq m\|x_k - x_{k+1}\|^2, \tag{1.6}$$

for some $m > 0$ where k is the number of steps taken since the latest restart, and

$$\text{Restart is initiated if } \nabla f(x_k + z_{k+1})^T z_{k+1} > 0. \tag{1.7}$$

We should mention that although condition (1.6) depends on the algorithm iterates rather than arbitrary x, y as we will see in the examples later that it is possible to check this condition and see that it is weaker than strong convexity. We are going to show that under conditions (1.5),(1.6),(1.7) Algorithm (4) has linear convergence rate.

1.6.2 Convergence rate of Gradient mapping based restart is linear

Now let's start our analysis of Gradient mapping based restart. Observe that when there is no restart (from (1.7))

$$(x_k - x_{k-1})^T \nabla f(x_k) \leq 0,$$

then

$$\|\beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k)\|^2 \geq \|\beta_k(x_k - x_{k-1})\|^2 + \alpha_k^2 \|\nabla f(x_k)\|^2, \tag{1.8}$$

where the left hand side is the momentum at the *next* step $k+1$: $\|x_{k+1} - x_k\|^2$, if there is no restart in that step either. Now let k_s denote the **first** iteration where we restart:

$$(x_{k_s} - x_{k_s-1})^T \nabla f(x_{k_s}) \leq 0. \tag{1.9}$$

Assume that

$$c(f(x_0) - f(x^*)) = f(x_{k_s}) - f(x^*).$$

To show linear convergence it is sufficient to establish that c has an upper bound strictly smaller than 1 and k_s also has an upper bound since this essentially means that the convergence rate will be $O(c^{k/\hat{k_s}})$($\hat{k_s}$ is the upperbound on k_s).

In the rest of the analysis, for the sake of simplicity, we will fix $\alpha_k = \alpha$ and $\beta_k = \beta$. This actually leads to a stronger result since it means that even with constant coefficients we can achieve linear convergence.

Lemma 1 *For fixed α and β and $k \leq k_s$,*

$$\|x_k - x_{k-1}\| \geq \alpha \sqrt{\frac{1}{M}\left(f(x_{k_s}) - f(x^*)\right) \sum_{i=0}^{k-1} \beta^{2i}}.$$

Proof: When there is no restart we have

$$\gamma_k \equiv \|x_k - x_{k-1}\| = \|\beta_{k-1}(x_{k-1} - x_{k-2}) - \alpha_{k-1}\nabla f(x_{k-1})\|.$$

From (1.5) we know that at each step $k \leq k_s$,

$$\|\nabla f(x_k)\| \geq \sqrt{(f(x_{k_s}) - f(x^*))/M}.$$

Combining this with (1.8), we get

$$\gamma_k^2 \geq \beta^2 \gamma_{k-1}^2 + \frac{1}{M}\alpha^2 (f(x_{k_s}) - f(x^*)),$$

which yields the desired bound when combined with the fact that

$$\gamma_1 = \alpha \|\nabla f(x_0)\|. \tag{1.10}$$

■

Lemma 1 shows that the worst case of growth happens when the current gradient is almost perpendicular to the momentum vector (intuitively we would expect that when these two vectors are almost perpendicular in the next step momentum vector will cause overshoot and should be zeroed out, see figure 1.6). This lemma actually gives us a very nice bound on how much the momentum will grow at each step as seen in Corollary 1.6.2. This will be a key ingredient in the proof of complexity.

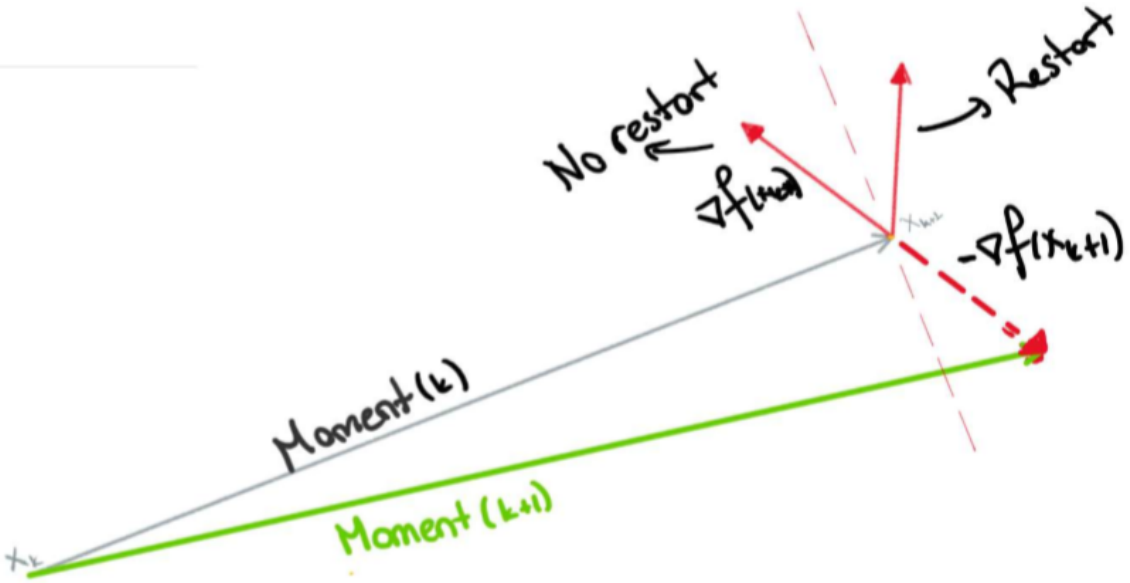


Figure 1.6: Evolution of Momentum

Corollary 1 *The momentum grows like* $O(\sqrt{k})$.

Proof: As β gets closer to 1 the momentum term will increase and from Lemma 1 the result follows. ■

Lemma 2 *Let* k_s *be the first restarting step. Then,*

$$f(x_0) - f(x^*) \geq (f(x_{k_s}) - f(x^*)) \left(1 + \frac{m}{M}\alpha^2 \sum_{k=0}^{k_s-1} \sum_{i=0}^{k} \beta^{2i}\right).$$

Proof: From (1.6), for $k < k_s$, and the fact that

$$f(x_k) - f(x^*) > f(x_{k_s}) - f(x^*),$$

we have,

$$f(x_k) - f(x_{k+1}) \geq \frac{m}{M}\alpha^2 (f(x_{k_s}) - f(x^*)) \sum_{i=0}^{k} \beta^{2i}.$$

Therefore

$$f(x_0) - f(x_{k_s}) \geq \frac{m}{M}\alpha^2(f(x_{k_s}) - f(x^*)) \sum_{k=0}^{k_s-1} \sum_{i=0}^{k} \beta^{2i},$$

which yields the desired bound. ■

In lemma 2 we see the relationship of magnitude of errors between the beginning and the end of every restart. The bound shows that there has to be a decrease by a constant factor.

Lemma 3 *If* $0 < \beta < 1$

$$k_s \leq \frac{1}{2\ln\beta} \ln\left(1 - \frac{1-\beta^2}{\frac{m}{M}\alpha^2}\right)_+ - 1.$$

Proof: From inequalities (1.5), (1.8), and Lemma 1, for fixed α and β we have:

$$\begin{aligned} \|x_{k+1} - x_k\|^2 &= \|\beta(x_k - x_{k-1}) - \alpha\nabla f(x_k)\|_2^2 \\ &\geq \|\beta(x_k - x_{k-1})\|^2 + \frac{1}{M}\alpha^2(f(x_k) - f(x^*)) \\ &\geq \beta^{2k}\|x_1 - x_0\|^2 \\ &\geq \frac{1}{M}\alpha^2\beta^{2k}\left(f(x_0) - f(x^*)\right). \end{aligned} \tag{1.11}$$

Substituting (1.6) in (1.11) and summing over k we get,

$$f(x_0) - f(x^*) \geq f(x_0) - f(x_{k_s}) \geq \frac{m}{M}\alpha^2 \sum_{k=0}^{k_s} \beta^{2k}(f(x_0) - f(x^*))$$

Hence,

$$1 \geq \frac{m}{M}\alpha^2 \frac{1-\beta^{2(k_s+1)}}{1-\beta^2},$$

which yields, when $0 < \beta < 1$,

$$k_s \leq \frac{1}{2\ln\beta} \ln\left(1 - \frac{1-\beta^2}{\frac{m}{M}\alpha^2}\right)_+ - 1.$$

■

Lemma 3 gives the other essential piece of information we need for the convergence which is the upper bound on the number of iterations between consecutive restarts. If we didn't have such an upper bound increasing k_s would certainly violate our result. We should also note that this upper bound on k_s is probably not sharp but it will suffice for our purposes. Although upper bound of k_s seems to be depending on m/M (as we will see for the adaptive restart this expression is a function of μ) selection of α can be completely automatized independent of m/M (one obvious way is to update α as $\frac{\alpha}{2}$ when $k_s < 2$).

Lemma 4 *If $\alpha < 1/L$ then $k_s \geq 2$. Also, for any $t \geq 2$, there exists an $\alpha > 0$, such that $k_s \geq t$.*

Proof: Since ∇f is Lipschitz continuous

$$\|\nabla f(x) - \nabla f\left(x - \alpha\nabla f(x)\right)\| \leq L\alpha\|\nabla f(x)\|.$$

If $0 < \alpha < L^{-1}$ then

$$\nabla f(x)^T \nabla f(x - \alpha\nabla f(x)) \geq \nabla f(x)^T \left(\nabla f(x) - L\alpha\nabla f(x)\right) \geq 0.$$

Therefore $k_s \geq 2$ since the initial momentum is zero.

A similar, but more tedious, argument, shows that for all $t \geq 2$ there exists a small enough $\alpha > 0$ such that $k_s \geq t$. The basic idea is that for sufficiently small α the initial momentum can be kept as small as desired. Then the Lipschitz continuity is used as

above to show that the restart condition will not be satisfied. ■

In lemma 4 we show that between consequtive restarts the algorithms takes at least one step if the step size is small enough. This prevents the case where the algorithm gets stuck by constantly restarting without updating x. This is a result of derivative being L-Lipshitz.

Now we can tidy things up and combine bounds on the number of steps in between consecutive restarts. Let k_j denote the number of iterations between the jth and $j-1$th restarts. Based on Lemmas 3 and 4, once $0 < \alpha < L^{-1}$ is fixed, we can choose $0 < \beta < 1$, such that there exist constants p and q which guarantee that

$$2 \leq p \leq k_j \leq q < \infty.$$

Lemma 5 *Let r be the total number of iterations. Then*

$$f(x_r) - f(x^*) \leq (f(x_0) - f(x^*)) \left[\frac{1}{1 + \alpha^2 \frac{m}{M} \sum_{k=0}^{p-1} \sum_{i=0}^{k} \beta^{2i}} \right]^{\frac{r}{q}}.$$

Proof: Let $\hat{x}_j$ denote the point right at the beginning of the jth restart where $\hat{x}_0 = x_0$. From Lemma (2) and $k_j \geq p$, right at the beginning of the jth restart we have,

$$f(\hat{x}_{j-1}) - f(x^*) \geq (f(\hat{x}_j) - f(x^*)) \left(1 + \frac{m}{M} \alpha^2 \sum_{k=0}^{p-1} \sum_{i=0}^{k} \beta^{2i} \right).$$

If there are a total of N restarts until iteration r this inequality leads to,

$$(f(x_r) - f(x^*)) \leq (f(\hat{x}_0) - f(x^*)) \left[\frac{1}{1 + \alpha^2 \frac{m}{M} \sum_{k=0}^{p-1} \sum_{i=0}^{k} \beta^{2i}} \right]^{N}.$$

From $k_j \leq q$ we have $N \geq \frac{r}{q}$ combining with $\hat{x}_0 = x_0$ the result follows. ■

Lemma 5 connects the initial error to the error on step r. With this lemma we have

all the ingredients necessary to lay down our main result.

Theorem 1 *Convergence rate of Algorithm (4) is linear.*

Proof: The lower and upper bounds on p and q from Lemmas 4 and 3 combined with the result in Lemma 5 yields

$$(f(x_k) - f(x^*)) \leq (f(x_0) - f(x^*)) \left[\frac{1}{1+\alpha^2\frac{m}{M}(\beta^2+1)}\right]^{\frac{k}{\frac{1}{2\ln\beta}\ln\left(1-\frac{1-\beta^2}{\frac{m}{M}\alpha^2}\right)_+^{-1}}}$$

Let

$$0 < \tau = \left[\frac{1}{1+\alpha^2\frac{m}{M}(\beta^2+1)}\right]^{\frac{1}{\frac{1}{2\ln\beta}\ln\left(1-\frac{1-\beta^2}{\frac{m}{M}\alpha^2}\right)_+^{-1}}} < 1.$$

Then we see that Algorithm 1 converges like $O(\tau^k)$ which is linear as claimed. ■

Algorithm 5: Momentum accelerated gradient algorithm with gradient-mapping restart

```
Choose x_{-1} ∈ R^n
x_0 = x_{-1}
for k ≥ 0 do
  z_{k+1} = β_k(x_k − x_{k−1}) − α_k ∇f(x_k)
  if ∇f(x_k + z_{k+1})^T z_{k+1} > 0 then
    x_{k+1} = x_k − α_k ∇f(x_k)
  else
    x_{k+1} = x_k + z_{k+1}
  end if
end for
```

Now that we have established linear convergence under the criteria 1.5, 1.6 and 1.7 the next question is "does strong convexity satisfy these criteria?". We are going to address this next.

1.6.3 Convergence Rate of Adaptive Restart under Strong Convexity

Now, we show that under strong convexity the adaptive restart rule achieves linear convergence (this version of the algorithm is given in Algorithm (5)). We shall henceforth refer to Algorithm 2 alternatively as MAGR.

Previously we have shown that if the three criteria we have given in (1.5), (1.6) and (1.7) are satisfied then the algorithm has linear convergence. So it is enough to establish that these criteria ((1.5), (1.6) and (1.7)) are satisfied.

Criterion (1.7) is satisfied by definition of the adaptive restart rule.

Assuming that no restart was initiated,

$$(x_{k+1} - x_k)^T \nabla f(x_{k+1}) \leq 0.$$

Then from equation (1.2) we have that,

$$f(x_k) \geq f(x_{k+1}) + \nabla f(x_{k+1})^T (x_k - x_{k+1}) + (\mu/2)\|x_{k+1} - x_k\|^2,$$

which implies that,

$$f(x_k) - f(x_{k+1}) \geq (\mu/2)\|x_{k+1} - x_k\|^2. \tag{1.12}$$

So criterion (1.6) is satisfied for $m = \frac{\mu}{2}$. Strong convexity can be also used to bound the gradients at each step.

$$f(x^*) - f(x) - \nabla f(x)^T (x^* - x) \geq (\mu/2)\|x - x^*\|^2,$$

where x^* denotes the minimum of f, leading to,

$$\begin{aligned} f(x) - f(x^*) &\leq \nabla f(x)^T(x - x^*) - (\mu/2)\|x - x^*\|^2 \\ &\leq \|\nabla f(x)\|\|x - x^*\| - (\mu/2)\|x - x^*\|^2 \\ &= \frac{\|\nabla f(x)\|^2}{2\mu} - \left(\|x - x^*\|\sqrt{\frac{\mu}{2}} - \frac{\|\nabla f(x)\|}{\sqrt{2\mu}}\right)^2 \\ &\leq \frac{\|\nabla f(x)\|^2}{2\mu}. \end{aligned} \tag{1.13}$$

hence criterion (1.5) is satisfied for $M = \frac{1}{2\mu}$. As we can see all three criteria are satisfied for this case hence we conclude that "Adaptive Restart" scheme has linear convergence under strong convexity.

1.7 Extension to Accelerated Gradient update rule

One question that is frequently asked is if this analysis is still valid for Nesterov's "Accelerated Gradient" algorithm. Although Lyapunov like analysis is used for the analysis of the non-restarted algorithm, for the restarted one we don't need this to establish linear convergence. This is not to say that such an analysis is useless, it is simply good for analyzing what happens in between consecutive restarts(where Nesterov's step size selection is still optimum).

Now let's sketch how this analysis can be extended for the generalized accelerated update rule in Definition (3). In accelerated gradient method the intermediate gradient $\nabla f(y_k)$ is accumulated instead of $\nabla f(x_k)$ hence instead of $(x_k - x_{k-1})^T\nabla f(x_k)$ algorithm checks $(y_k - x_{k-1})^T\nabla f(y_k)$ and restarts if this inner product is positive. In this case instead of (1.8) we will get $\|\beta_k(x_k - x_{k-1}) - \alpha_k\nabla f(y_k)\|_2^2 \geq \|\beta_k(x_k - x_{k-1})\|^2 + \alpha_k^2\|\nabla f(y_k)\|^2$ and the rest of the analysis will follow very similarly. Since the gradient is smooth it is reasonable to expect that $(x_{k+1} - x_k)^T\nabla f(x_{k+1})$ and $(y_k - x_{k-1})^T\nabla f(y_k)$ to have the

same sign most of the time (it also gives almost identical results in the experiments) however using $(y_k - x_{k-1})^T \nabla f(y_k)$ as the restarting criteria prevents any complications. The choice of β_k as in Accelerated gradient method will be faster than constant β. Since $\beta_k \to 1$ monotonically, in the analysis when the constant step size as β is chosen greater than β_2 we guarantee that the summation $\sum_{k=0}^{k_s} \beta_k^{2k}$ stays small enough so that bounds for k_s still hold.

1.7.1 A Non-Strongly Convex example

A simple example of a non-strongly convex function that satisfies the two conditions is $f(x) = x^T Ax/2$ where A is a symmetric positive semi-definite matrix with at least one zero eigenvalue. Since A is not full rank it is obvious that this objective function is not strongly convex.

However at every x that is not a minimum we have

$$\nabla f(x) = Ax = \sum_i c_i v_i \neq 0,$$

where v_i are eigenvectors corresponding to eigenvalues $\lambda_i > 0$ (Since A is positive semi definite, the eigenvectors corresponding to 0 eigenvalue span the null-space of A hence $\nabla f(x)$ is a linear combination of eigenvectors corresponding to non-zero eigenvalues). Then for $\hat{\lambda} = \min_{\lambda_i > 0} \lambda_i$ we have

$$f(x) - f(x^*) = \frac{x^T Ax}{2} \leq \frac{x^T A^2 x}{2\hat{\lambda}} = \frac{\|\nabla f(x)\|^2}{2\hat{\lambda}}.$$

So, criterion (1.5) is satisfied. Since $x_k - x_{k+1}$ is a linear combination of the gradients until step k then it is a linear combination of eigenvectors corresponding to non-zero

eigenvalues. We get

$$(x_k - x_{k+1})^T A(x_k - x_{k+1}) \geq \hat{\lambda}\|x_k - x_{k+1}\|^2.,$$

and

$$x_k^T A x_k - x_{k+1}^T A x_{k+1} = (x_k - x_{k+1})^T A(x_k - x_{k+1}) + 2(x_k - x_{k+1})^T A x_{k+1}.$$

From the restart condition $\nabla f(x_{k+1})^T(x_k - x_{k+1}) \geq 0$, we conclude that,

$$x_k^T A x_k - x_{k+1}^T A x_{k+1} \geq (x_k - x_{k+1})^T A(x_k - x_{k+1}) \geq \hat{\lambda}\|x_k - x_{k+1}\|^2,$$

which satisfies criterion (1.6). Therefore, this example is not strongly convex yet it satisfies both criteria and adaptive restart scheme will achieve linear convergence.

1.8 Cone based restart

We now introduce a new gradient based restart criteria which we call "cone based restart". As we will see in the experiments the corresponding Algorithm (6) has very similar convergence behavior and speed to Algorithm (5), but it has some nice properties that make it easier to analyze. Moreover the coefficient c in Algorithm (6) makes it possible to tune the algorithm.

The restart condition

$$\nabla f(x_k + z_{k+1})^T g_r < c\,\|\nabla f(x_k + z_{k+1})\|\|g_r\|,$$

guarantees that all of the gradients until the next restart lie in the cone centered around a

Algorithm 6: Momentum accelerated gradient algorithm with cone based restart

Choose $x_{-1} \in R^n$
Choose $c > 1/\sqrt{2}$
$x_0 = x_{-1}$
$g_r = \nabla f(x_0)$
for $k \geq 0$ **do**
  $z_{k+1} = \beta_k(x_k - x_{k-1}) - \alpha_k \nabla f(x_k)$
  if $\nabla f(x_k + z_{k+1})^T g_r < c\|\nabla f(x_k + z_{k+1})\|\|g_r\|$ **then**
    $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
    $g_r = \nabla f(x_{k+1})$
  else
    $x_{k+1} = x_k + z_{k+1}$
  end if
end for

pivot vector (gradient right after the restart) g_r. For strongly convex objective functions it is easy to see that they indeed are satisfied since the selection $c > \frac{1}{\sqrt{2}}$ guarantees that for all k we have $(x_k - x_{k+1})^T \nabla f(x_{k+1}) > 0$ when there is no restart.

1.8.1 Another Example

We will look into the following problem.

$$f(x) = \rho \log \left(\sum_{i=1}^{m} \exp((a_i^T x - b_i)/\rho) \right). \tag{1.14}$$

This problem is a smoothed version of the more general problem of

$$f(x) = \max_{i=1,\ldots,m} (a_i^T x - b_i). \tag{1.15}$$

In our numerical experiments we took $\alpha_k = 0.99$ and $\beta_k = (r+1)/(r-1)$ where r is the number of steps taken after the latest restart.

The experiments (see figures 1.10 and 1.11)show linear convergence of both Cone

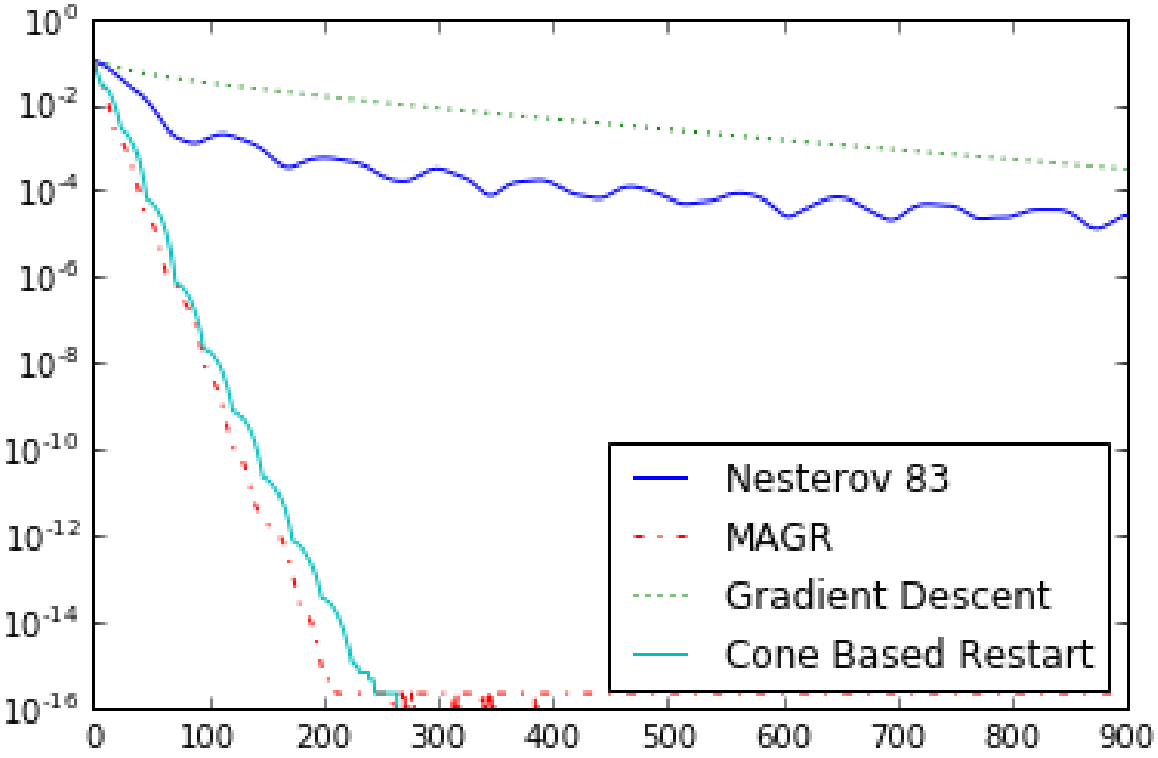


Figure 1.7: Optimizing the smooth version for $\rho = 1$. The vertical axis depicts $\frac{(f(x_n)-f^*)}{f^*}$, and the horizontal axis depicts the iteration number n.

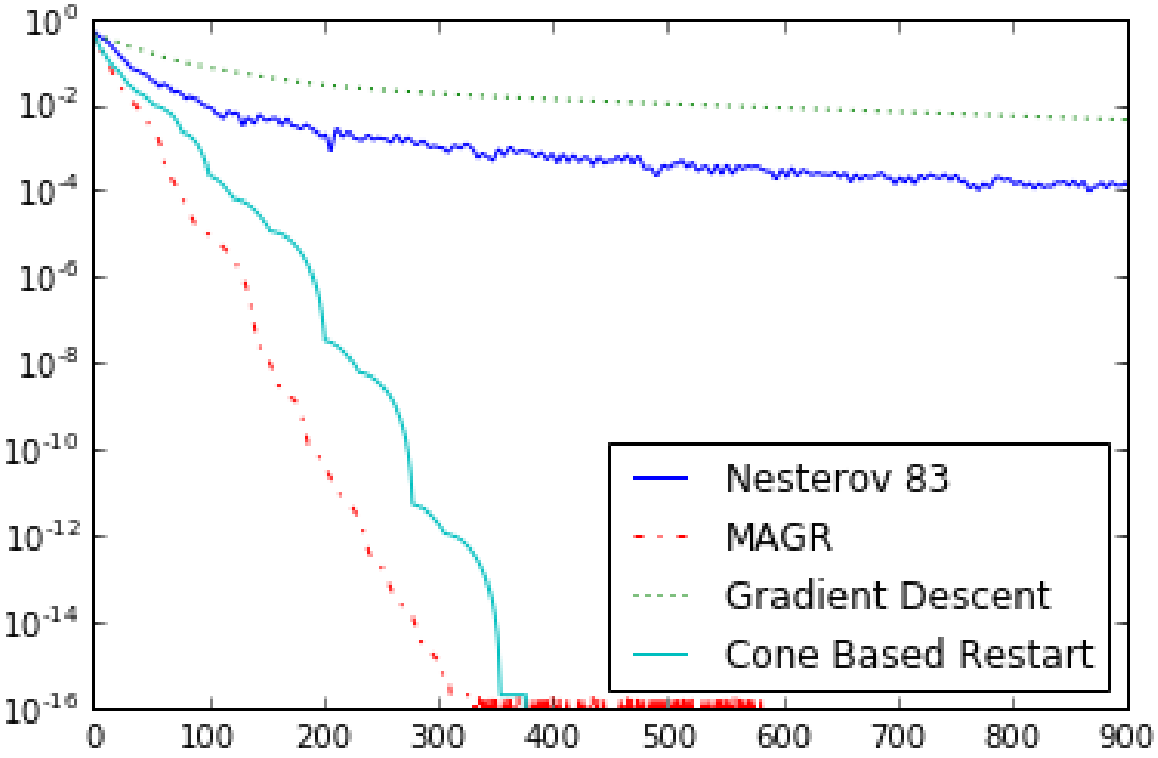


Figure 1.8: Optimizing the smooth version for $\rho = 0.1$. The vertical axis depicts $\frac{(f(x_n)-f^*)}{f^*}$, and the horizontal axis depicts the iteration number n.

Based Restart and Algorithm (5), and how close their convergence behaviors are.

1.9 An algorithm for Non-smooth functions

In this section we are going to be looking into the case where the objective function is non-smooth. The usual approach to solve these problems is to solve an approximate smooth version of the objective(which is what we did in the previous subsection). We on the other hand, will give an extension of the MAGR which can be used in non-smooth convex functions(We will call it NSMAGR(7)).

Algorithm 7: Non-Smooth Momentum accelerated gradient algorithm with gradient-mapping restart

Choose $x_{-1} \in R^n$
Choose $\mu \in (0, 1)$
$x_0 = x_{-1}$
for $k \geq 0$ **do**
  Choose $g_k \in \partial f(x_k)$
  $z_{k+1} = \beta_k(x_k - x_{k-1}) - \alpha_k g_k$
  Choose $\hat{g} \in \partial f(x_k + z_{k+1})$
  if $\hat{g}^T z_{k+1} > 0$ **then**
    if $\hat{g}^T g_k < 0$ **then**
      $\beta_{k+1} = \mu\beta_k$
      $x_{k+1} = x_k + z_{k+1}$
    else
      $x_{k+1} = x_k - \alpha_k g_k$
      $\beta_{k+1} = \beta_k$
    end if
  else
    $x_{k+1} = x_k + z_{k+1}$
  end if
end for

In the algorithm NSMAGR, g_k and $\hat{g}$ are sub-gradients of the objective function at x_k and $x_k + z_{k+1}$ respectively. As with any gradient based non-smooth algorithm we are using sub-gradients instead of gradients. The main difference NSMAGR and MAGR

is that we impose a second condition for restart which is the algorithm will not restart if there is an abrupt change in the gradient. To check if there is an abrupt change we compute the inner product of the sub-gradients at the current and the projected next step, if it is negative we simply don't restart instead decrease the step-size since it is leading to too much of a decrease. In the algorithm, we have taken $\beta_k = \mu\beta_k$ however one can select different reduction schemes as long as it does not hinder convergence rate.

1.9.1 A Non Smooth Example

We return to the problem in subsection 1.8.1 as we will see this function is actually a smooth approximation of the function in 1.16 (We will be trying our non-smooth version of MAGR on minimizing this function) which is given as follows.

$$\text{minimize } max_{i=1\dots m}(a_i^T x - b_i) \tag{1.16}$$

Although smooth approximation converges to the original function as $\rho \to 0$ it will cause numerical problems as it becomes too small. In the experiments $\mu = 0.99$ and $\alpha_k = (r+1)/(r-1)$ where r is the number of steps taken after the latest restart.

One can observe that for all values of $/rho$ the accelerated gradient with restart is drastically better than both vanilla accelerated gradient and gradient descent. For larger values classic accelerated gradient is asymptotically exceeded by gradient descent yet MAGR stays the fastest. For the non-smooth case however the results are very encouraging. While the algorithms are no longer monotonic, the decrease rate of the NSMAGR is still linear(figure 1.12) yet bot accelerated gradient and the vanilla gradient descent get very slow and can not get close to the optimum.

If we look at the minimum value achieved up to each step(figure 1.13) we can see this better.

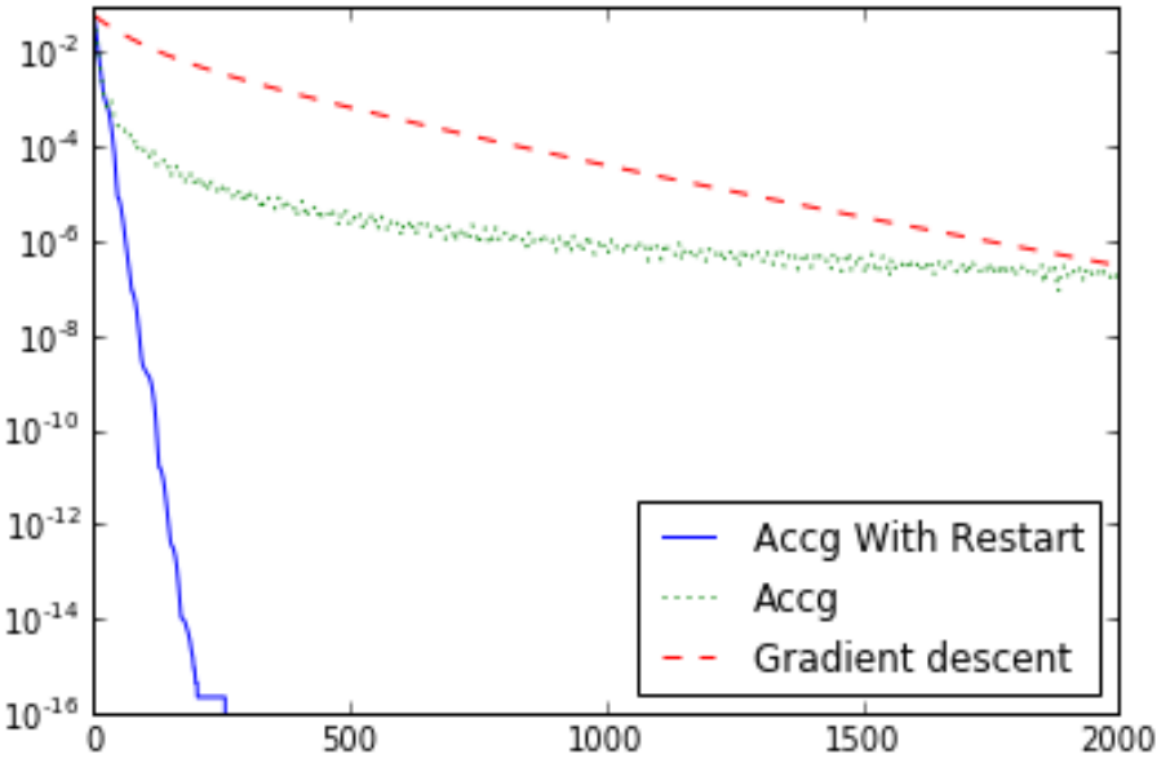


Figure 1.9:

Figure 1.10: Optimizin the smooth version for $\rho = 1$

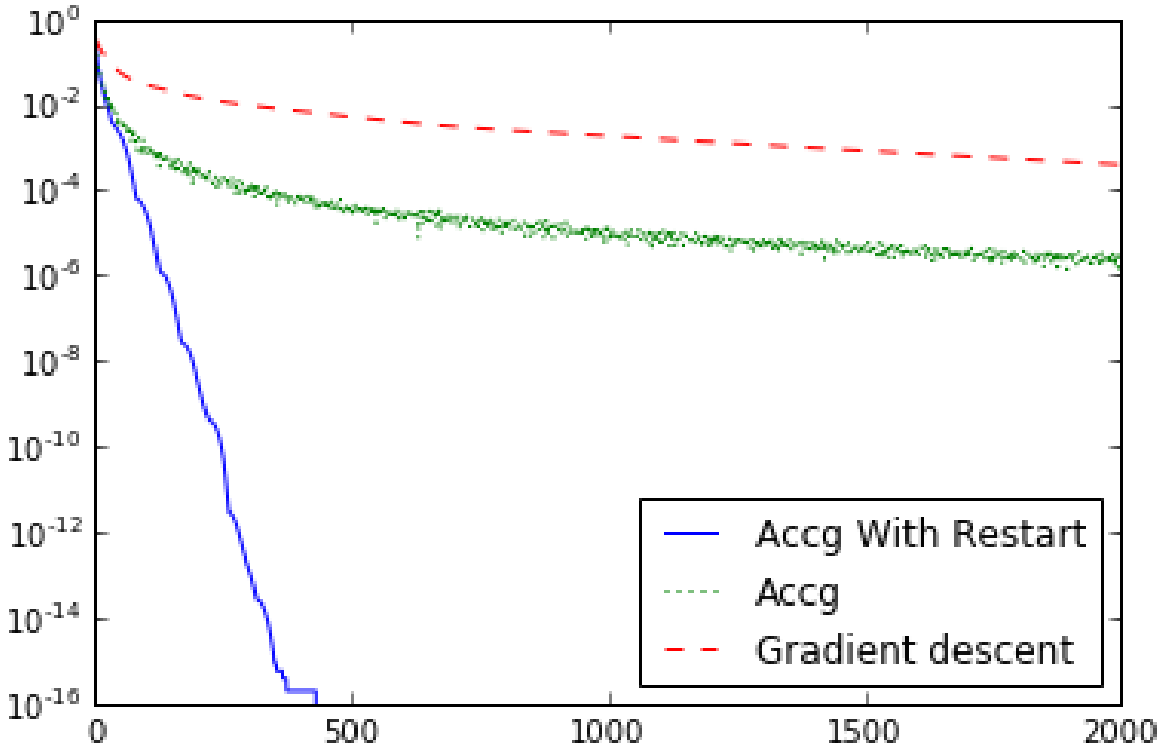


Figure 1.11: Optimizin the smooth version for $\rho = 0.1$

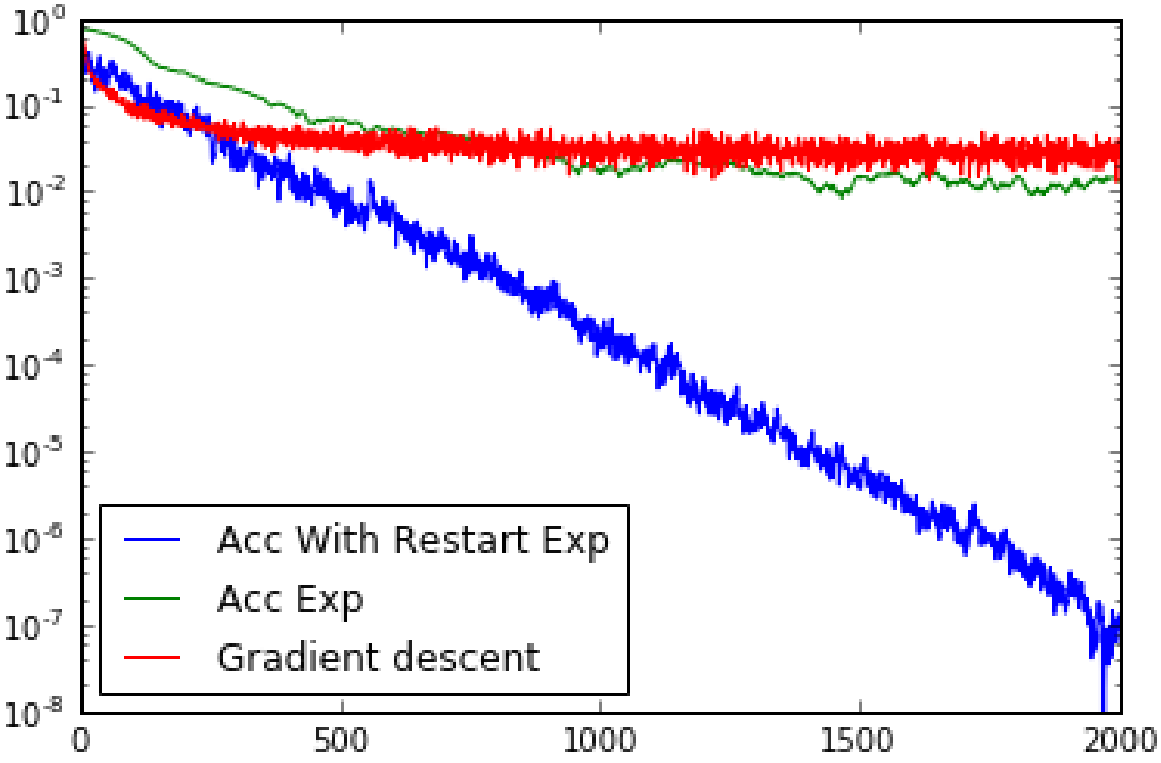


Figure 1.12: Non-Smooth problem minimization

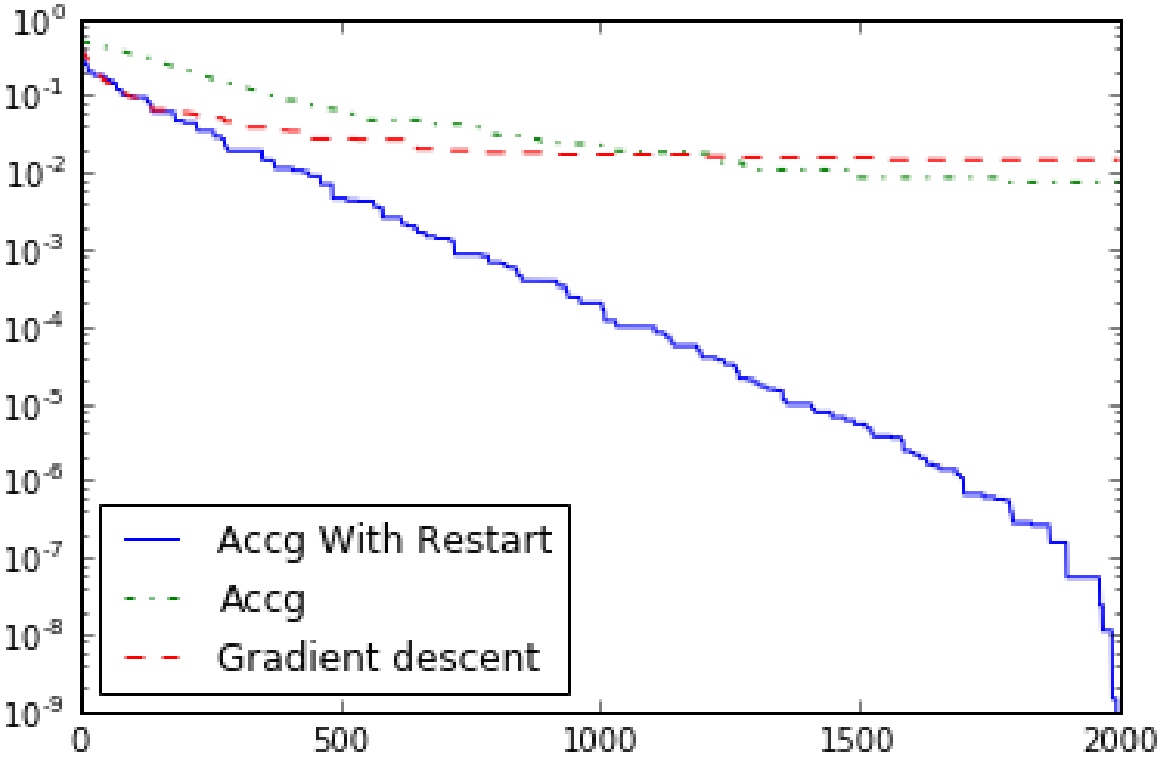


Figure 1.13: Minimums achieved at each step for non-smooth problem

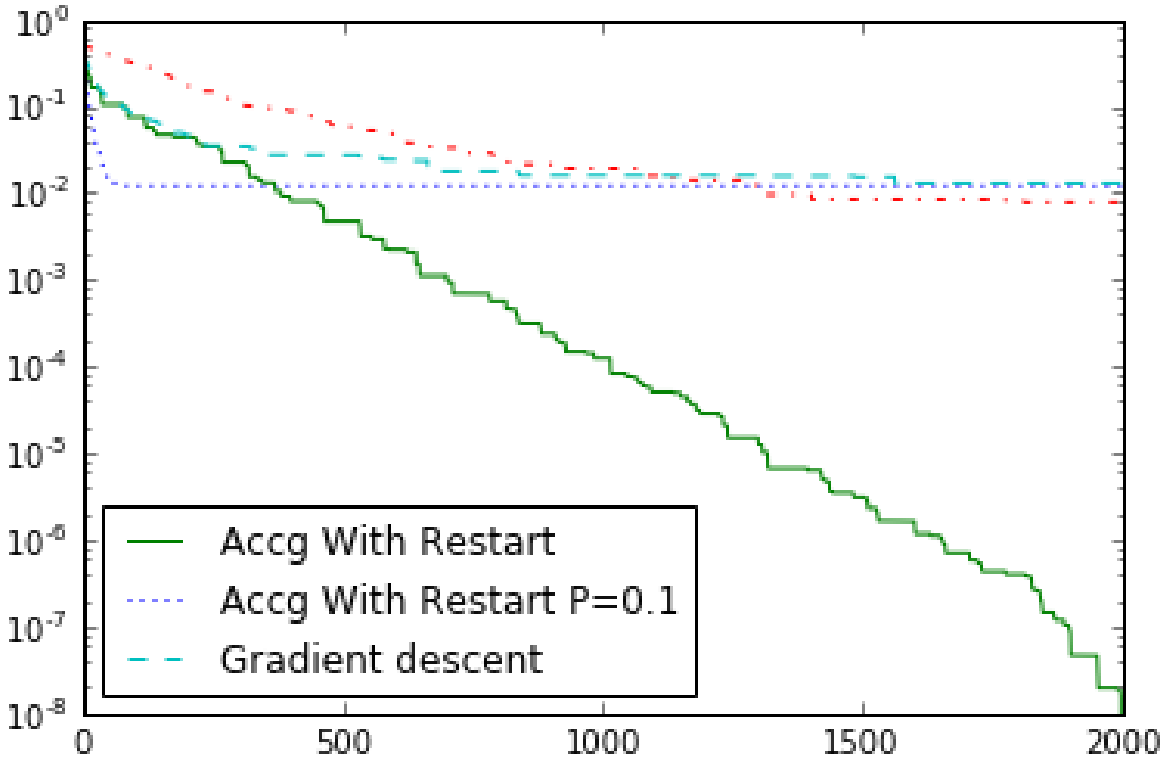


Figure 1.14: NSMAGR vs MAGR on $\rho = 0.1$

We also do a comparison on how fast and accurate NSMAGR is compared to MAGR for the smooth case. In figure 1.14 we can see that although MAGR on smooth problem is fast in the beginning it converges to a point which is not close to the optimum while NSMAGR gets very close to the optimum. Also one has to note that using the smooth function creates a constant overhead at each step so in fact NSMAGR is also way faster in terms of flop count.

1.10 Conclusions

Recent analysis of accelerated gradient methods have been based on ODEs [3, 6]. The rough idea is to analyze the continuous case, where step size is arbitrarily small,

and then expand the analysis by quantizing the continuous path. Here however we used the classical approach in proving the convergence rate. With the restart condition the algorithm becomes monotonic. The momentum vector in the worst case grows like $O(\sqrt{k})$, and even in this case we can prove linear convergence rate. For experimental results on how effective this restart rule is, the reader can refer to [4, 5].

In this chapter we have shown that the gradient-mapping based restart scheme will improve the convergence rate of momentum based algorithms to linear. Although this was suspected to be the case in practice we have now proved it to be true under the assumptions of strong convexity and smoothness. Both of these seem to be essential ingredients since it is easy to give examples of non-smooth functions where restart actually worsens the convergence rate (becomes comparable to that of standard gradient descent). However reusing earlier gradients seems to be capable of resolving this issue and will be reported elsewhere.

Chapter 2

Sparsest Solution to an Underdetermined System of Linear Equations via Penalized Huber Loss

2.1 Introduction

The purpose of this chapter paper is to prove an extrapolation property to calculate (the) sparsest solution to an underdetermined linear system by approximating the Basis Pursuit (BP) model, and to explore ties between the sparsest solution, the L1-norm solution and the Huber loss function (c.f. [12]). The problem of interest is posed as:

$$\min_x \{\|x\|_0 : Ax = b\}, \tag{2.1}$$

where $\|x\|_0$ counts the non-zero elements of the vector $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ $(m < n)$ assumed to be a full rank matrix (with rank equal to m), and $b \in \mathbb{R}^m$. The problem has numerous applications in signal processing, compressed sensing, error decoding, image

denoising and so on. Since it is NP-hard, one may seek to solve an approximate problem instead of (2.1). A popular choice is the Basis Pursuit method [13]. The Basis Pursuit approach to sparse recovery (there is ample literature on the subject, which cannot be reviewed within this brief paper; see e.g., [14, 15, 16, 17], or [18, 19] for in-depth monographs on the problem of compressive sensing), and is based on the following problem referred to as [SL1]:

$$\min_x \{\|x\|_1 : Ax = b\}. \tag{2.2}$$

In addition to the commonly observed fact that the problem (2.2) usually gives a sparsest solution to the linear system in numerical calculation, and with high probability according to several theoretical results in literature, it was proved that (2.2) is the Lagrange bi-dual of (2.1); c.f., [20]. An alternative view to duality for (2.1) can be found in [21].

Let f^* denote the optimal value of (2.2) and X_* denote its optimal solution set. The dual problem is

$$\max_\xi \{b^T\xi : \|A^T\xi\|_\infty \leq 1\}. \tag{2.3}$$

Denote the optimal set of the dual problem as Ξ^0. The notation s^0 refers to the sign vector with components the usual sign function denoted sgn in $\{0, \pm 1\}$ such that

$$\text{sgn}(t) = \begin{cases} 0 & \text{if } t = 0 \\ 1 & \text{if } t > 0 \\ -1 & \text{if } t < 0. \end{cases}$$

The diagonal matrix $W^0(.)$ is obtained from $s^0(.)$ using $W^0_{ii} = 1 - (s^0_i)^2$ for $i = 1, \ldots, n$. Optimality conditions for (2.2) can be summarized in the following result (see e.g., [22]) which is convenient for our purposes.

Proposition 1 *Let $x \in \mathbb{R}^n$ with $s = s^0(x)$ and the associated $W = W^0(x)$. Then, x*

solves (2.2) if and only if there exists $\lambda \in \mathbb{R}^m$ *and* $d \in \mathbb{R}^n$ *such that* $\|Wd\|_\infty \leq 1$ *and* $A^T\lambda = Wd + s$.

Clearly, if the minimizer x is unique in (2.2) the matrix $\begin{pmatrix} A \\ W \end{pmatrix}$ has full rank.

Define the Huber loss function depending on a tuning constant $\gamma > 0$:

$$\rho(t) = \begin{cases} \frac{1}{2\gamma}t^2, & \text{if } |t| \leq \gamma \\ |t| - \frac{\gamma}{2}, & \text{otherwise.} \end{cases}$$

The Huber function is an example of a convex quadratic spline (a piecewise quadratic); see e.g., [23] for background on convex quadratic splines. In a previous paper [24] the second author investigated necessary and sufficient conditions for exact recovery of an individual sparse vector using the Huber function instead of the L1 function, i.e., using the problem

$$\min_x \{\sum_{i=1}^{n} \rho(x_i) | Ax = Au\}$$

where u is some sparse real vector. The approach of the present paper is different since we shall consider the problem

$$\min_x \{\frac{1}{2\gamma}\|b - Ax\|_2^2 + \Phi_\gamma(x)\}, \tag{2.4}$$

where $\Phi_\gamma(x) = \sum_{i=1}^{n} \rho(x_i)$, as an approximation to (2.2) for smaller and smaller values of γ. We shall refer to the problem as penalized Huber Loss as the first term acts a simple penalty function. Problems that are reminiscent of (2.4) were studied in the context of linear and quadratic programming in [25, 26, 27]. Reference [22] is also close in spirit to the present paper in that it utilizes quadratic programming problem similar to ours for computing a sparsest solution although neither Huber loss function nor extrapolation

ideas are present in [22]. Another reference related to the present is [28] where the Huber function is used within the context of a min-max concave penalty and saddle point computations for least squares regularization.

The problem (2.4) aims to achieve two goals at the same time by pushing γ to zero: recover a solution to the system $Ax = b$ in the limit while the Huber loss function collapses to the 1-norm. It may be thought that the method advocated here will cause numerical problems as one seems to have to push γ to extremely small values to obtain an approximate result. However, the crux of the results of the paper is that one does not need to deal with very small values of γ due to a convenient extrapolation result described in section II. A variant of the extrapolation property using extreme point minimizers is discussed in section III. Section IV is devoted to further exploration of the ties between a sparsest solution, the unique solution to problem (2.2), and the penalized Huber Loss problem (2.4).

2.2 An Extrapolation Procedure

The first (to be expected) result is that the solutions of (2.4) tend to the solutions of (2.2) as γ tends to zero.

Proposition 2 *The sequence of solutions $\{x^\gamma\}$ of (2.4) approaches X_* as γ tends to zero: $\forall \epsilon > 0$, $\exists \gamma(\epsilon)$ such that for $\gamma \leq \gamma(\epsilon)$, x^γ solves (2.4) $\implies$ dist$(x^\gamma, X_*) = \min_{x \in X_*} \|x^\gamma - x\| \leq \epsilon$.*

Proof: Let x_0 be a solution to (2.2). For ease of notation, define $f_\gamma(x) = \frac{1}{2\gamma}\|b - Ax\|_2^2 + \Phi_\gamma(x)$. We have for all $x \in \mathbb{R}^n$:

$$0 \leq \|x\|_1 - \Phi_\gamma(x) \leq n\frac{\gamma}{2}. \tag{2.5}$$

Let x^i denote a minimizer of f_γ. Then one has

$$\|x^i\|_1 \leq \|x^i\|_1 + \frac{1}{2\gamma}\|b - Ax^i\|_2^2 \leq \Phi_\gamma(x^i) + n\frac{\gamma}{2} + \frac{1}{2\gamma}\|b - Ax^i\|_2^2. \tag{2.6}$$

On the other hand we have that

$$\Phi_\gamma(x^i) + \frac{1}{2\gamma}\|b - Ax^i\|_2^2 \leq f^* \tag{2.7}$$

because

$$\Phi_\gamma(x^i) + \frac{1}{2\gamma}\|b - Ax^i\|_2^2 \leq \Phi_\gamma(x_0) + \frac{1}{2\gamma}\|b - Ax_0\|_2^2 \leq \|x_0\|_1 = f^*. \tag{2.8}$$

Using (2.7) in (2.6) we get

$$\|x^i\|_1 \leq f^* + n\frac{\gamma}{2}. \tag{2.9}$$

Now, let $x^i \to \bar{x}$, if necessary after passing to a subsequence. Then $\bar{x}$ must satisfy $Ax = b$ since otherwise we have

$$\lim_{i\to\infty, \gamma_i \searrow 0} \Phi_\gamma(x^i) + \frac{1}{2\gamma}\|b - Ax^i\|_2^2 = \|\bar{x}\|_1 + \lim_{i\to\infty} \frac{1}{2\gamma}\|b - Ax^i\|_2^2$$

and

$$\lim_{i\to\infty, \gamma_i \searrow 0} \frac{1}{2\gamma}\|b - Ax^i\|_2^2 = +\infty.$$

This leads to a contradiction with (2.9). So $\bar{x}$ satisfies $A\bar{x} = b$. By (2.6) one has

$$\|\bar{x}\|_1 = \lim_{i\to\infty} \|x^i\|_1 \leq f^*.$$

Hence $\bar{x} \in X_*$. ■

While the above gives a convergence result in the limit, we shall not have to let γ tend to zero. We shall instead devise a simple extrapolation procedure to get to a solution of (2.2) from a solution of (2.4) under a sign constancy condition.

The optimality conditions of (2.4) can be expressed in a compact notation as follows; define $s^\gamma \in \{0, \pm 1\}^n$ as follows:

$$s_i^\gamma(t) = \begin{cases} 0 & \text{if } |t| \leq \gamma \\ 1 & \text{if } t > \gamma \\ -1 & \text{if } t < -\gamma \end{cases}$$

with the diagonal matrix $W^\gamma(.)$ derived from $s^\gamma(.)$ using $W_{ii}^\gamma = 1 - (s_i^\gamma)^2$ for $i = 1, \ldots, n$. Then, the following result can be stated.

Lemma 1 *Let $\bar{x} \in \mathbb{R}^n$. Then $\bar{x}$ solves (2.4) if and only if the following equation holds:*

$$\frac{1}{\gamma} W^\gamma(\bar{x})\bar{x} + s^\gamma(\bar{x}) - \frac{1}{\gamma} A^T(b - A\bar{x}) = 0. \tag{2.10}$$

We shall obtain an extrapolation procedure based on Lemma 1. Let us rewrite equation (2.10) for a minimizer x^γ as

$$(A^T A + W^\gamma(x^\gamma))x^\gamma = A^T b - \gamma s^\gamma(x^\gamma). \tag{2.11}$$

Corollary 1 *Let x^γ solve (2.4) uniquely. Then* $\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix}$ *has full rank.*

Proof: If $\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix}$ does not have full rank, there exists $h \neq 0$ such that

$\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix} h = 0$, which implies that there exists $\delta > 0$ such that $s^\gamma(x^\gamma + \delta h) = s^\gamma(x^\gamma)$.

■

Lemma 2 *Let x^γ solve (2.4). Then the system of linear equations*

$$(A^T A + W^\gamma(x^\gamma))d = s^\gamma(x^\gamma) \tag{2.12}$$

is consistent.

Proof: By rearranging (2.11) we obtain

$$-\frac{1}{\gamma}(A^T\ W^\gamma(x^\gamma)) \begin{pmatrix} r^\gamma \\ x^\gamma \end{pmatrix} = s^\gamma(x^\gamma),$$

where $r^\gamma \equiv Ax^\gamma - b$. Then (2.12) is equivalent to

$$(A^T\ W^\gamma(x^\gamma)) \begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix} d = -\frac{1}{\gamma}(A^T\ W^\gamma(x^\gamma)) \begin{pmatrix} r^\gamma \\ x^\gamma \end{pmatrix},$$

which is consistent since it is the normal equations for the system

$$\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix} h = \begin{pmatrix} r^\gamma \\ x^\gamma \end{pmatrix}.$$

■

For ease of notation let $W = W^\gamma(x^\gamma)$ and $s = s^\gamma(x^\gamma)$. Let the SVD of $\begin{pmatrix} A \\ W \end{pmatrix} = U\Sigma V^T$ where $U \in \mathbb{R}^{(m+n)\times(m+n)}$, $\Sigma \in \mathbb{R}^{(m+n)\times n}$, $V \in \mathbb{R}^{n\times n}$ with $U^T U = I_{(m+n)}$ and $VV^T = V^T V = I_n$. Then we have $A^T b = \sum_{j=1}^n \alpha_j v_j$ where v_js are columns of V, and

$s = \sum_{j=1}^{n} \beta_j v_j$, for some scalars α_js and β_js. More compactly, we have $A^T y - \gamma s = V\alpha - \gamma V\beta$.

Now, let us re-write (2.11) as

$$VS^2V^T x^\gamma = V\alpha - \gamma V\beta$$

where $S^2 = \Sigma^T\Sigma$ is $n \times n$ diagonal matrix with diagonal entries σ_j^2, $j = 1, \ldots, n$ (σ_j are singular values of $\begin{pmatrix} A \\ W \end{pmatrix}$), α is the vector of α_js, and idem for the vector β of β_js. If $\begin{pmatrix} A \\ W \end{pmatrix}$ has full rank, then S^2 is invertible, i.e., $\sigma_j > 0$ for all $j = 1, \ldots, n$. Therefore, an expression for x^γ is obtained as

$$x^\gamma = \sum_{j=1}^{n} \frac{\alpha_j - \gamma\beta_j}{\sigma_j^2} v_j. \tag{2.13}$$

Now, if $s^\gamma(x^\gamma)$ remains constant at some s for $\gamma \in (0, \gamma^*]$, one has

$$\lim_{\gamma \to 0} x^\gamma = \sum_{j=1}^{n} \frac{\alpha_j}{\sigma_j^2} v_j \equiv x^\#.$$

Now, consider the linear system of equations

$$(A^T A + W)d = s$$

for $\gamma \in (0, \gamma^*]$ (i.e., the range of γ where the sign vector remains constant). By the

analysis above and under the full rank assumption on $\begin{pmatrix} A \\ W \end{pmatrix}$, one has

$$d = \sum_{j=1}^{n} \frac{\beta_j}{\sigma_j^2} v_j.$$

Hence, using the expression (2.13) for x^γ gives

$$x^\gamma + \gamma d = x^\#.$$

Combining the above with Proposition 1, one obtains the result below. Let X^γ denote the set of minimizers of (2.4).

Proposition 3 *If X^γ is a singleton and $s^\gamma(x^\gamma)$ remains constant for $\gamma \in (0, \gamma^*]$, and then*

$$x^\gamma + \gamma d = x^0,$$

where d solves

$$(A^T A + W^\gamma(x^\gamma))d = s^\gamma(x^\gamma),$$

and x^0 solves (2.2).

Proof: If X^γ is a singleton for sufficiently small $\gamma > 0$ then the $\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix}$ has full rank. Then using the sign constancy, the result follows from the previous analysis, Corollary 1 and Proposition 1. ■

Proposition 3 implies that if for $\gamma > 0$ sufficiently small one has uniqueness of the minimizer and the sign constancy condition satisfied then one can zoom along to a solution of problem (2.2) from a solution of (2.4) after solving a system of linear equations. Note that we used SVD only for the sake of analysis. In actual computation one does not need

to compute the SVD. A more economical factorization of $\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix}$ (e.g., QR) can be used.

Before exploring further the property of sign constancy and ties between the solutions to (2.2), (2.4) and the sparsest solution, we shall first weaken the uniqueness condition on x^γ in Proposition 3. To do this we need to study the dual problem to (2.4). First, let us observe that the dual problem to (2.4) is the following strictly concave quadratic optimization problem:

$$\max_{\xi}\{-\frac{\gamma}{2}\xi^T(I_m + AA^T)\xi + b^T\xi : \|A^T\xi\|_\infty \leq 1\}, \tag{2.14}$$

with a unique solution, ξ^γ. Optimality conditions for the pair of primal-dual problems imply the following relations between a solution x^γ to (2.4) and the dual solution ξ^γ:

$$A^T\xi^\gamma = \frac{1}{\gamma}W^\gamma(x^\gamma)x^\gamma + s^\gamma(x^\gamma), \tag{2.15}$$

$$\frac{1}{\gamma}\xi^\gamma = b - Ax^\gamma. \tag{2.16}$$

Furthermore, by a classical result of [29], for sufficiently small $\gamma > 0$, we have that $\xi^\gamma = \xi^*$ is the least weighted-norm solution of the dual problem (2.3) using the norm term $\|\begin{pmatrix} I \\ A^T \end{pmatrix}x\|_2$ after viewing the dual problem above as

$$\max_{\xi}\{-\frac{\gamma}{2}\|\begin{pmatrix} I \\ A^T \end{pmatrix}\xi\|_2^2 + b^T\xi : \|A^T\xi\|_\infty \leq 1\}.$$

I.e., ξ^* solves the problem

$$\min_{\xi \in \Xi^0} \|(\begin{matrix} I \\ A^T \end{matrix})\xi\|_2.$$

2.3 Extreme Points and Extrapolation

Now recall that x is an extreme point of X^γ if and only if $x = \lambda y + (1-\lambda)z$ implies $x = y = z$ for any $\lambda \in (0,1)$.

Lemma 3 *x is an extreme point of x^γ if and only if $\begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix}$ has full rank.*

Proof: We proceed as in [30]. Suppose that $x \in X^\gamma$ and $\begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix}$ has rank less than m. Then there exists a vector $u \in \mathbb{R}^n$, not identically zero, such that $\begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix} u = 0$. Now we can pick an $\epsilon > 0$ sufficiently small such that $(x \pm \epsilon u)_i > \gamma$ for all i such that $x_i > \gamma$ and $(x \pm \epsilon u)_i < -\gamma$ for all i such that $x_i < -\gamma$. Observe that $A(x \pm \epsilon u) = Ax$ and $W^\gamma(x)(x \pm \epsilon u) = W^\gamma(x)x$. Thus, we have $s^\gamma(x \pm \epsilon u) = s^\gamma(x)$. Hence, both $x \pm \epsilon u$ satisfy (2.10) and we get that both $x \pm \epsilon u \in X^\gamma$. Since $x = \frac{1}{2}(x+\epsilon u) + \frac{1}{2}(x - \epsilon u)$ with $u \neq 0$, x is not an extreme point of X^γ.

Suppose $x \in X^\gamma$ and $\begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix}$ has rank m. If $y, z \in X^\gamma$ are such that $x = \lambda y + (1-\lambda)z$ for some $\lambda \in (0,1)$ then by (2.15) we have $\gamma(A^T\xi^\gamma)_i = x_i$ if and only if $y_i = z_i = \gamma(A^T\xi^\gamma)_i$. Thus we have $W^\gamma(x)x = W^\gamma(x)y = W^\gamma(x)z$. By (2.16) we have $\gamma(b - Ax)_i = \xi_i^\gamma$ if and only if $\gamma(b - Ay)_i = \xi_i^\gamma = \gamma(b - Az)_i$, which implies $Ax = Ay = Az$. Hence we have $\begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix} x = \begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix} y = \begin{pmatrix} A \\ W^\gamma(x) \end{pmatrix} z$, which in turn implies that $x = y = z$. Therefore, x is an extreme point of X^γ. ■

Using the above lemma and the analysis leading to Proposition 3, we can state directly the extrapolation property even in the absence of uniqueness of the minimizer, provided the sign vector associated with an an extreme point minimizer remains constant for sufficiently small γ.

Proposition 4 *Let x^γ be an extreme point minimizer of (2.4) for $\gamma \in (0, \gamma^*]$. If $s^\gamma(x^\gamma)$ remains constant for $\gamma \in (0, \gamma^*]$, then*

$$x^\gamma + \gamma d = x^0,$$

where d solves

$$(A^T A + W^\gamma(x^\gamma))d = s^\gamma(x^\gamma),$$

and x^0 solves (2.2).

It turns out that we can weaken the conditions used in the analysis of the present section even further.

Lemma 4 *For every instance of the problem there exists a sign vector s and corresponding W where for all γ there is a $\gamma' \leq \gamma$ and a corresponding solution $x^{\gamma'}$ such that $s^{\gamma'}(x^{\gamma'}) = s$ and $W^{\gamma'} = W$.*

Proof: Each sign vector $s \in \{0, \pm 1\}^n$ so the set of possible sign vectors, denote it by S, is finite (likewise set of possible W matrices is also finite). Assume the contrary, let γ_s be such that for $\gamma < \gamma_s$ and for all solutions x^γ we have $s(x^\gamma) \neq s$. Since set S is finite then for $\gamma < \bar{\gamma} = \min_{s \in S} \gamma_s$ no sign vector can survive. However $\forall \gamma \leq \bar{\gamma}$ we have a solution set and for every extreme point x^γ of this solution set we have $s(x^\gamma) \in S$ hence a contradiction. ■

This lemma shows that even if the minimizer of (2.4) is not unique there exists a sign

vector and W matrix corresponding to an extreme point of the solution set for arbitrarily small γ. Then we can relax Proposition 3 as follows.

Proposition 5 *For sufficiently small γ^* for every $\gamma \leq \gamma^*$ any extreme point of X^γ will satisfy*

$$x^\gamma + \gamma d = x^0,$$

where d solves

$$(A^T A + W^\gamma(x^\gamma))d = s^\gamma(x^\gamma),$$

and x^0 solves (2.2).

Proof: For every extreme point of X^γ the corresponding $\begin{pmatrix} A \\ W^\gamma(x^\gamma) \end{pmatrix}$ has full rank. Moreover for small enough γ^* only the infinitely repeating sign vectors remain that is if $s = s(x_{\gamma^*})$ where x^{γ^*} is an extreme point of X^{γ^*} then for all $\gamma \leq \gamma^*$ there exists a $\gamma' \leq \gamma$ such that $s = s(x^{\gamma'})$. The proof then follows from analysis of Proposition (3). ■

2.4 Sign Constancy and Ties Among Solutions

Now, we shall turn to the question of sign constancy and ties among various solutions.

Definition 1 *A dual optimal solution to (2.3) is called non-degenerate if there exists a primal optimal solution x^* such that the following conditions hold:*

$$(A^T\xi^*)_i = 1 \implies x_i^* > 0,$$

$$(A^T\xi^*)_i = -1 \implies x_i^* < 0,$$

$$|(A^T\xi^*)_i| < 1 \implies x_i^* = 0.$$

For any sign vector, we define $\kappa_s^+ = \{i : s_i = 1\}$, and $\kappa_s^- = \{i : s_i = -1\}$ with $\kappa_s = \kappa_s^+ \cup \kappa_s^-$, and $\kappa_s^0 = \{i : s_i = 0\}$.

Proposition 6 *Let x_0 be a unique solution to (2.2) with $\mathbf{s}_0 = s(x_0)$ and x^γ be the unique minimizer of (2.4) with dual solution ξ^* non-degenerate for sufficiently small $\gamma > 0$. Then, there exists γ^* such that $s^\gamma(x^\gamma)$ remains constant for $\gamma \in (0, \gamma^*]$.*

Proof: Since for sufficiently small $\gamma > 0$ the unique $x^\gamma \to x_0$ by Theorem 1, we have $|x_i^\gamma| > \gamma$ for $i \in \kappa_{\mathbf{s}_0}$ and sufficiently small $\gamma > 0$. Because $\xi^* \in \Xi^0$ for sufficiently small $\gamma > 0$ and by the non-degerenacy assumption, we have from (2.15) that $|x_i^\gamma|/\gamma < 1$ for $i \in \kappa_{\mathbf{s}}^0$ and γ sufficiently small. ■

For the next result, we use $s^0 \in \{0, \pm 1\}^n$ defined before with the diagonal matrix $W^0(.)$ derived from $s^0(.)$ using $W_{ii}^0 = 1 - (s_i^0)^2$ for $i = 1, \ldots, n$, as before.

Proposition 7 *Let x_0 be a (sparse) solution of $Ax = b$ with $s = s^0(x_0)$ and $W = W^0(x_0)$. If there exists a solution d to the system of linear equations:*

$$(A^T A + W)d = s \tag{2.17}$$

with the property

$$\|Wd\|_\infty \leq 1, \tag{2.18}$$

then there exists γ^* such that

$$s^\gamma(x_0 - \gamma d) = s$$

and $x_0 - \gamma d$ solves (2.4) for $\gamma \in (0, \gamma^*]$. Furthermore, x_0 solves (2.2).

Proof: We can re-write (2.17) as

$$(A^T A + W)d = \frac{1}{\gamma}(A^T A + W)x_0 - \frac{1}{\gamma}A^T b + s$$

for some $\gamma > 0$ since $Ax_0 = b$ and $Wx_0 = 0$. By simple algebra, rearranging the previous we obtain

$$(A^TA + W)(x_0 - \gamma d) = A^Tb - \gamma s \tag{2.19}$$

Now, let $\delta = \min\{(x_0)_i : |(x_0)_i| \neq 0\}$. Choose $0 < \gamma_2 < \delta$ such that for $\gamma \in (0, \gamma_2]$ one has

$$(x_0 - \gamma d)_i > \gamma_2 \text{ for } i \in \kappa_s^+,$$

$$(x_0 - \gamma d)_i < -\gamma_2 \text{ for } i \in \kappa_s^-.$$

Combining the above with (2.18), we have that $s^\gamma(x_0 - \gamma d) = s$, and using (2.19) $x_0 - \gamma d$ solves (2.4) for $\gamma \in (0, \gamma^*]$ where $\gamma^* = \gamma_2$.

On the other hand, the fact that there exists a solution d to (2.17) with $\|Wd\|_\infty \leq 1$ implies that one can take $\lambda \equiv Ad$. Hence, one has $A^T\lambda = -Wd + s$. But $\|Wd\|_\infty = \| - Wd\|_\infty$. Hence x_0 solves (2.2) by Proposition 1. ■

If x_0 of the previous result also satisfies a slight regularity condition in addition to the one already presented, then an immediate corollary is obtained as follows.

Corollary 2 *Let x_0 be a (sparse) solution of $Ax = b$ with $s = s^0(x_0)$ and $W = W^0(x_0)$ such that $\begin{pmatrix} A \\ W \end{pmatrix}$ has full rank. If the unique solution d to the system of linear equations:*

$$(A^TA + W)d = s \tag{2.20}$$

has the property

$$\|Wd\|_\infty \leq 1, \tag{2.21}$$

then there exists γ^* such that

$$s^\gamma(x_0 - \gamma d) = s$$

and $x_0 - \gamma d$ solves (2.4) for $\gamma \in (0, \gamma^*]$. Furthermore, x_0 solves (2.2).

In particular, the above two results (Proposition 5 and Corollary 2) are valid for a sparsest solution to the linear system of equations. Hence, they constitute sufficient conditions for the sparsest solution to be a minimizer of (2.2).

A more substantial sign constancy result is proved below relaxing some of the conditions imposed previously (i.e., non-degeneracy) in the present section.

Theorem 1 *Let x_0 be a unique solution to (2.2) with $s_0 = s(x_0)$. If x^γ solving (2.4) is unique for sufficiently small γ then there exists γ^* such that $s^\gamma(x^\gamma)$ remains constant for $\gamma \in (0, \gamma^*]$ with $\kappa^+_{s_0} \subseteq \kappa^+_{s^\gamma}$, $\kappa^-_{s_0} \subseteq \kappa^-_{s^\gamma}$.*

Proof: Let x_0 be the unique solution to (2.2) with $\mathbf{s}_0 = s(x_0)$. Since the number of different sign vectors s is finite, there must a exist a sequence of positive numbers $\gamma_1, \gamma_2, \ldots$, with $\gamma_k \searrow 0$ for $k \to \infty$ such that $s^\gamma(x^\gamma)$ is constant for $\gamma = \gamma_k$ for $k = 1, 2, \ldots$. Denote this sign vector by $\mathbf{s}$. Let $\mathcal{D} = \{i : |(A^T\xi^\gamma)_i| = 1\} \cap \kappa_{\mathbf{s}^0}$, and $\mathcal{S} = \{s \in \hat{S} : s_i = \mathbf{s}_{0i} \text{ for } i \notin \mathcal{D}\}$. Since $x^\gamma \to x_0$ by Theorem 1, we have $|x^\gamma_i| > \gamma$ for $i \in \kappa_{\mathbf{s}_0}$ and sufficiently small $\gamma > 0$. Since $\xi^\gamma \in \Xi^0$ for sufficiently small $\gamma > 0$, we have from (2.15) that $|x^\gamma_i|/\gamma < 1$ for $i \in \kappa^0_{\mathbf{s}_0} \setminus \mathcal{D}$ and γ sufficiently small. Therefore,we must have that $s \in \mathcal{S}$ since $\gamma_k \searrow 0$.

Now, consider $\mathbf{s}$ (the sign vector that is encountered infinitely many times as defined above) and its associated diagonal matrix $\mathbf{W}$. Let the SVD factorization of $\begin{pmatrix} A \\ \mathbf{W} \end{pmatrix} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and consider the system

$$(A^TA + \mathbf{W})d = \mathbf{s}. \tag{2.22}$$

This system is consistent by Lemma 2 since x_γ satisfies (2.11). We have, as previously,

$$d_\gamma = \sum_{j=1}^{n} \frac{\beta_j}{\sigma_j^2} \mathbf{v}_j$$

if $\begin{pmatrix} A \\ \mathbf{W} \end{pmatrix}$ has full rank. Since $\psi_i(\gamma) = -\gamma(d_\gamma)_i$ is a rational function of γ, it can only have a finite number of oscillations as $\gamma \searrow 0$; hence there exists $\gamma_1^* > 0$ such that either $|\psi_i(\gamma)| > 1$ for $\gamma \in (0, \gamma_1^*]$ or $|\psi_i(\gamma)| \leq 1$ $\gamma \in (0, \gamma_1^*]$. If $i \notin \kappa_{\mathbf{s}_0}^0$ then $(x_0)_i = 0$ and $(x_0)_i - \gamma(d_\gamma)_i = \psi_i(\gamma)$. Hence, the ith component of $s^\gamma(x_0 - \gamma d_\gamma)$ is constant for $\gamma \in (0, \gamma_1^*]$. Since d_γ is bounded, the other components of $s^\gamma(x_0 - \gamma d_\gamma)$ must also be constant in some interval $(0, \gamma_2^*]$. Therefore, $s^\gamma(x_0 - \gamma d_\gamma)$ is constant for $\gamma \in (0, \gamma_3^*]$ where $\gamma_3^* = \min\{\gamma_1^*, \gamma_2^*\}$.

Now, let $\gamma = \gamma_k \in (0, \gamma_3^*]$ be a value for which $s^\gamma(x^\gamma) = \mathbf{s}$. It follows from the above development that the unique minimizer x^γ is given as $x_0 - \gamma d$. ■

We can now merge the results of Proposition 3 with Theorem 1 into the following.

Theorem 2 *Let x_0 be a unique solution to (2.2) with $s_0 = s(x_0)$. If x^γ solving (2.4) is unique for sufficiently small γ, then*

$$x^\gamma + \gamma d = x_0,$$

where d solves

$$(A^T A + W^\gamma(x^\gamma))d = s^\gamma(x^\gamma).$$

We have made the assumption that both the solution x_0 to (2.2) and a minimizer x^γ, for $\gamma > 0$ sufficiently small, to (2.4) are unique. A legitimate question is whether the uniqueness of x_0 implies that of x^γ. The answer is negative as the following example

shows.

Example 1 *Let*

$$A = \begin{pmatrix} 1 & -1 & 1/2 & -1 \\ 0 & 0 & -17 & 2 \end{pmatrix},$$

with $b = (-1\ 2)^T$. *The unique solution to (2.2) is* $x^0 = (0\ 0\ 0\ 1)^T$. *For sufficiently small* $\gamma \in (0, 0.195]$, *the set of minimizers of (2.4) is the interval between the extreme points*

$$(-\gamma\ \ 2\gamma\ \ -\gamma/2\ \ 1-17\gamma/4)^T$$

and

$$(-2\gamma\ \ \gamma\ \ -\gamma/2\ \ 1-17\gamma/4)^T.$$

We can weaken the uniqueness condition of Theorem 1 as shown in the next result.

Proposition 8 *Let* x_0 *be a unique solution to (2.2) with* $s = s^0(x_0)$ *and* $W = W^0(x_0)$ *and let* $\bar{s}$ *(and* $\bar{W}$ *derived from* $\bar{s}$*) be a sign vector such that* $\kappa_{\bar{s}}^0 \subseteq \kappa_s^0$ *with* $\kappa_s^+ \subseteq \kappa_{\bar{s}}^+$, *and* $\kappa_s^- \subseteq \kappa_{\bar{s}}^-$ *with* $\begin{pmatrix} A \\ \bar{W} \end{pmatrix}$ *having full rank. If the unique solution* d^* *to*

$$(A^T A + \bar{W})d = \bar{s} \tag{2.23}$$

has the properties $\|\bar{W}d^*\|_\infty \leq 1$, *and for all* i *such that* $s_i = 0$ *and* $\bar{s}_i \neq 0$ *it holds that* $\bar{s}_i d_i^* < 0$ *and* $|d_i^*| > \max_{j:s_j=\bar{s}_j=0} |d_j^*|$ *then there exists* γ^* *such that*

$$s^\gamma(x_0 - \gamma d) = \bar{s}$$

and $x_0 - \gamma d$ *is an extreme point minimizer of (2.4) for* $\gamma \in (0, \gamma^*]$.

Proof: Similar to the proof of Proposition 7, we re-write (2.23) as

$$(A^TA + \bar{W})d^* = \frac{1}{\gamma}(A^TA + \bar{W})x_0 - \frac{1}{\gamma}A^Tb + \bar{s}$$

for some $\gamma > 0$ since $Ax_0 = b$ and $\bar{W}x_0 = 0$. Rearranging the previous we obtain

$$(A^TA + \bar{W})(x_0 - \gamma d^*) = A^Tb - \gamma\bar{s}. \tag{2.24}$$

Now, let $\delta = \min\{(x_0)_i : |(x_0)_i| \neq 0\}$. Choose $0 < \gamma_2 < \delta$ such that for $\gamma \in (0, \gamma_2]$ one has

$$(x_0 - \gamma d^*)_i > \gamma_2 \text{ for } i \in \kappa_{\bar{s}}^+,$$

$$(x_0 - \gamma d^*)_i < -\gamma_2 \text{ for } i \in \kappa_{\bar{s}}^-,$$

due to the conditions imposed on d^*. Combining the above with (2.18), we have that $s^\gamma(x_0 - \gamma d) = \bar{s}$, and using (2.24) $x_0 - \gamma d$ solves (2.4) for $\gamma \in (0, \gamma^*]$ where $\gamma^* = \gamma_2$, and is an extreme point minimizer. ■

Example 2 *Notice that all conditions of the previous proposition are fulfilled in Example 1. If one takes* $\bar{s} = (0\ 1\ 0\ 1)^T$ *then one gets* $d^* = (1\ \ -2\ 1/2\ 17/4)$*, and for* γ *small enough, one obtains the extreme point minimizers* $(-\gamma\ \ 2\gamma\ \ -\gamma/2\ \ 1 - 17\gamma/4)^T$*. On the other hand, if one takes* $\bar{s} = (-1\ 0\ 0\ 1)^T$*, one gets the* $d^* = (2\ \ -1\ 1/2\ 17/4)$ *and recovers the extreme points* $(-2\gamma\ \ \gamma\ \ -\gamma/2\ \ 1 - 17\gamma/4)^T$ *for sufficiently small* γ*.*

To conclude the paper, we give a necessary condition for recovery of a sparsest solution; a similar result is also given in [24] for exact recovery of an individual sparse vector. The necessary condition gives an unexpected relationship between the sparsest solution recovery (by recovery it is meant that the sparsest solution should be unique in (2.2)) and the unique solution to (2.14) being the unique solution to (2.3) as well.

Theorem 3 *Let $\hat{x}$ be a sparsest solution to $Ax = b$ with $\|\hat{x}\|_0 < m$. If $\hat{x}$ solves (2.2) uniquely then for sufficiently small $\gamma > 0$ the unique solution to (2.14) is not the unique solution to (2.3).*

Proof: For sufficiently small $\gamma > 0$ the unique solution to (2.14) is the least norm solution of (2.3). By the theory of linear programming (c.f. [31]), the dual problem (2.3) admits an extreme point optimal solution ξ^* with the set $\mathcal{A}(\xi^*) = \{i : |(A^T\xi^*)_i| = 1\}$ having at least m elements. If ξ^* is the unique optimal solution to (2.3) (it is the unique optimal solution to (2.14)) then by the Goldman-Tucker strict complementarity theorem (c.f. [32]) and the uniqueness of $\hat{x}$, the support of $\hat{x}$ should have cardinality at least m, which is a contradiction. ■

2.5 Conclusion and Future directions

In this chapter we have introduced "penalized Huber-loss" approach to solve the problem of finding the sparsest solution to a system of underdetermined set of linear equations. We further showed that the sign constancy condition is not necessary and one can work with the extreme points to reach the same result. Finding the optimum of the "penalized Huber-Loss" function and solving the linear system afterwards are both computationally cheap hence we are hoping that these results will lead to fast method for finding the sparsest solution. A future research direction is to analyze the case where σ is taken close to machine precision and see if the problem is computationally tractable with convergence guarantees.

Chapter 3

Equilibrium on Constrained Trade Networks

3.1 Introduction

In this final chapter where which is a work in progress, we venture into a topic which has strong ties to both optimization and Economics. Trades are a fundamental part of Economies. The basic definition of a trade is buying/selling goods, services etc. We are interested in trade networks ([33]) where the trades are based bilateral contracts(a contract between a single buyer and seller). In this model every edge corresponds to a trade, nodes represent agents and the orientation of the edge identifies "buyer" and "seller". The agents have quasi-linear utilities with associated prices on them. Each agent can be both a buyer and a seller(which lets us model real life networks accurately). The goods/services which are indivisible. This leads to edge flows, which represent the quantity of the trade, being integer valued.

In this model we are interested in the existence of competitive equilibrium(prices that for the quantities of edge flows which maximizes each agent's utility clears the mar-

ket). In ([33]) it was established that if agents' preferences satisfy full substitutability condition competitive equilibrium exists. Full substitutability is a generalization of gross-substitutes property introduced by Kelso-Crawford([34]). Gross substitute property is proven to be equivalent to valuations being $M^\natural$ concave (Fujishige-Yang, [35],Murota-Tamura[36]).

In this chapter we are going to give results on our ongoing work. We assume that agent's utilities are quasi-lineaer with non-linear part being $M^\natural$ concave. In the previous work we have mentioned so far, the model only considers independent capacity constraints on each edge(contract). We investigate the existence of Competitive equilibrium under more complex constraints over the contracts.

3.1.1 Contributions

Our contributions so far are as follows. We first give a counterexample which demonstrates that if the feasible set of link flows is a polymatroid then there may not exists a Competitive equilibrium. We then prove that if the set of feasible net flows is a polymatroid then we have an integral optima to the SMFP with complex constraints and that may lead to existence of Competitive Equilibrium. We finally focus our attention to the case where for each node the set of feasible outgoing/incoming link flows is a polymatroid and we give an example of a trade network formulation which guarantees Competitive Equilibrium.

3.2 Discrete Convexity

Although we are going to be focusing on M-convexity we should start from Discrete convexity which is in some sense the extension of the concept of convexity to the discrete domain. For a comprehensive treatment of the subject the reader should refer to Murota

[37]. The first question is "what qualifies as a "convex" function in the discrete domain?". A discrete function $f : Z^n \to R$ is said to be convex extensible if there exists a convex function $\hat{f} : R^n \to R$ such that for all $x \in Z^n$ we have $f(x) = \hat{f}(x)$. This definition seems like a natural way of defining convexity for the discrete domains however it fails to capture the desirable properties of Convexity such as

- Any local minimum being a global minimum
- Separating hyperplane theorem
- Duality

. These properties require locality conditions on the functions which are much stronger than just convex-extensibility (except the 1-dimensional domain in which case these properties carry over trivially).

3.2.1 Integral Convexity

We can write the convex-extension more formally as follows(the intersection of the supporting hyperplanes of the epigraph).

$$\hat{f}(x) = min\{\sum_z \lambda_z f(z) \| \sum_z \lambda_z z = x, 0 \leq \lambda_z \leq 1, \sum_z \lambda_z = 1\}$$

As we said earlier we need a stronger locality condition. We will start by restricting our attention to the nearest neighbours of every point.

Definition 6 $\tilde{f}(x) = min\{\sum_{z \in N(x)} \lambda_z f(z) \| \sum_{z \in N(x)} \lambda_z z = x, 0 \leq \lambda_z \leq 1, \sum_z \lambda_z = 1\}$ *will be called the* ***convex envelope*** *of function* f *where* $N(x) = \{z \| -1 < x_i - z_i < 1, \forall i\}$

Convex envelope is the continuous extension of the function where we are just using convex combinations of each point's immediate integral neighbors' function values defined by $N(x)$) to create the envelope function. One can observe that while convex extension always led to a convex function this convex envelope need not.
We are going to call a function integrally-convex if its convex-envelope is also convex(it will be equal to its convex extension).

Definition 7 *A discrete function $f : Z^n \rightarrow R$ is **integrally convex** if its convex-envelope is convex.*

Integral convexity is much stronger and for all integrally convex functions local minimum has to be a global minimum. Although this is very promising integral convexity itself still may not always be enough for creating fast algorithms since the neighbors of a point will still increase exponentially w.r.t. the dimension of the domain.
The set definition can be similarly given as follows.

Definition 8 *A set M is called an integrally convex set if there exist an integrally convex indicator function f such that $M = \{x \| f(x) = 1\}$*

The figure (3.1) from (Moriguchi [38]) shows the difference between convex and integrally convex sets nicely.

3.2.2 L-Convexity

A stronger notion then integral convexity is L-convexity. The idea is the approximation of the mid point between two points which may or may not be an integer.

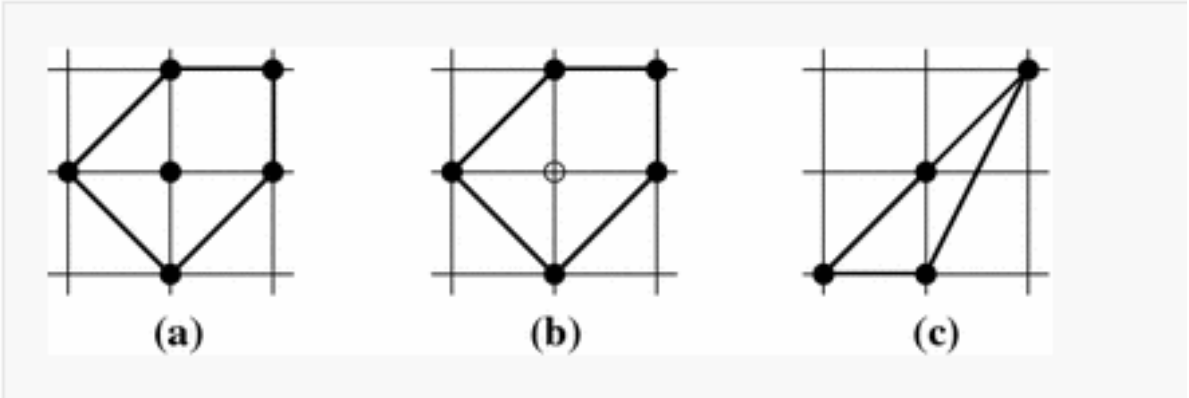


Figure 3.1: a is integrally convex, b and c are not

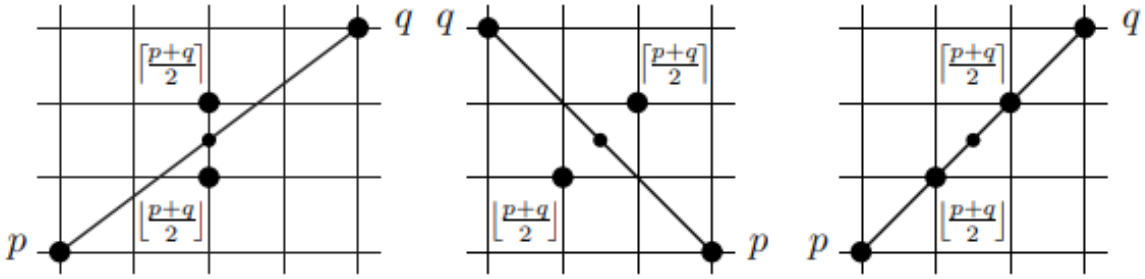


Figure 3.2: Midpoint Convexity (Murota)

Definition 9 *A discrete function $f : Z^n \to R$ is called* ***L-convex*** *if for all $p, q \in Z^n$ it satisfies the following.*

$$f(p) + f(q) \geq f(\lfloor \frac{p+q}{2} \rfloor) + f(\lceil \frac{p+q}{2} \rceil)$$

In figure 3.2 the idea behind the Midpoint convexity is demonstrated where the average of two points is compared to the approximation of the mid point.

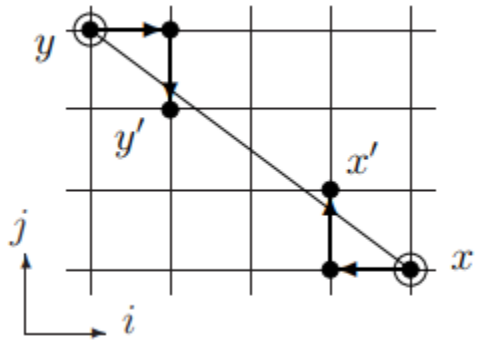


Figure 3.3: A simple exchange between two points x and y

3.3 M-Convexity

After introducing some other locality conditions we arrive at the condition that we will use in this chapter which is $M - convexity$. Our definitions here are due to Kazuo Murota.

For all $\delta \in Z^n$ we define $supp^+(\delta) = \{i \| \delta_i > 0\}$ and $supp^-(\delta) = \{i \| \delta_i < 0\}$. The notion of $M - convexity$ is based on the exchanges between the sets $supp^+$ and $supp^-$. Let's start by defining the "M-convex" set.

Definition 10 *A set $B \subseteq Z^n$ is called "M-convex" if for all $x, y \in B$ and foll all $u \in supp^+(x - y)$ there exists a $v \in supp^-(x - y)$ such that $x - \chi_u + \chi_v$ and $y + \chi_u - \chi_v$ are both in the set B.*

In definition (10) χ_u refers to the basis vector where uth element is one and all others are zero. A sample exchange can be seen in figure (3.3).

M-convex functions are defined similarly in (11).

Definition 11 *A function $f : Z^n \to R$ for an M-convex domain B is called to be M-convex if for all $x, y \in B$, $\forall u \in supp^+(x - y)$ there exists a $v \in supp^-(x - y)$ such*

that

$$f(x) + f(y) \geq f(x - \chi_u + \chi_v) + f(y + \chi_u - \chi_v).$$

To see the connection of M-convex function to convex functions in the classical sense we should consider an arbitrary convex function f ,two points x, y in its domain and some $\lambda \in [0, 1]$. By definition

$$\lambda f(x) + (1 - \lambda) f(y) \geq f(\lambda x + (1 - \lambda) y)$$

and

$$\lambda f(y) + (1 - \lambda) f(x) \geq f(\lambda y + (1 - \lambda) x).$$

When we sum them up we get

$$f(x) + f(y) \geq f(\lambda x + (1 - \lambda) y) + f(\lambda y + (1 - \lambda) x).$$

We switch the points on the right hand side with the exchanges which leads to the formulation in definition 11. This is somewhat an approximation of a point on the line segment connecting the points x and y (taking two integer points equal distance from the line segment and connecting them). M-convexity has strong ties with submodularity. A submodular set function f is one that satisfies $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq E$.

Theorem 2 *For any M-convex set B the function defined as $f(S) = max\{x(S) \| x \in B\}$ for every $S \subseteq E$ is a submodular function.*

Proof: For all $S, T \subseteq B$ take $x, y \in B$ such that $x(S \cap T) = f(S \cap T)$ and $y(S \cup T) = f(S \cup T)$ and having minimum $\|x - y\|_1$. For such choice of x, y we have for all $i \in S \cap T$ $x(i) = y(i)$, otherwise there is an exchange between x and y for some $u \in S \cap T$

that is going to lead to an x' where $x'(S \cap T) \geq x(S \cap T)$ and $\|x-y\|_1 > \|x'-y\|_1$. Then $f(S \cap T)+f(S \cup T) = x(S \cap T)+y(S \cup T) = y(S \cap T)+y(S \cup T) = y(S)+y(T) \leq f(S)+f(T)$.

■

Theorem 2 gives a connection between M-convex functions and integral submodular set functions. The inverse is also correct(corollary 2).

Corollary 2 *If f is an integer valued submodular function then there exists an M-convex set B such that $f(S) = max\{x(S) \| x \in B\}$.*

3.3.1 $M^\natural - Convexity$

The definition of M-convexity relies on the exchange between two different elements yet this can be weakened to cover single element exchanges. We start with the set definition first.

Definition 12 *A set $B \subseteq Z^n$ is called "$M^\natural$-convex" if for all $x, y \in B$ and foll all $u \in supp^+(x-y)$ we have one of the following.*

- *$x - \chi_u$ and $y + \chi_u$ are both in B*
- *there exists a $v \in supp^-(x-y)$ such that $x - \chi_u + \chi_v$ and $y + \chi_u - \chi_v$ are both in the set B.*

$M^\natural - convex$ functions are defined as follows.

Definition 13 *A function $f : Z^n \to R$ is called to be $M^\natural$-convex if for all $x, y \in B$, $\forall u \in supp^+(x-y)$ we have one of the following.*

- *$f(x) + f(y) \geq f(x - \chi_u) + f(y + \chi_u)$*
- *there exists a $v \in supp^-(x-y)$ such that $f(x)+f(y) \geq f(x-\chi_u+\chi_v)+f(y+\chi_u-\chi_v)$.*

The domain of a function $f : Z^n \to R$ is defined as $domf = \{x \in Z^n \| f(x) < \infty\}$. The consequence of a function being $M^\natural$-convex is

Lemma 6 *The domain of an $M^\natural$-convex is an $M^\natural$-convex.*

Some examples of $M^\natural$-convex. On a M^{sharp}-convex domain the following functions are $M^\natural$-convex.

$$f(x) = a + bx$$

$$f(x) = \sum_i a_i x_i^2, a_i \geq 0$$

3.3.2 Some Properties and Remarks on M/$M^\natural$ Convex functions

Let's start by giving equivalent definitions of M-convex functions. The netxt lemma is due to Murota [39].

Lemma 7 *(Lemma 2.1, [39])For $B \subseteq Z^n$ the following three conditions are equivalent.*

- *For distinct x, y in B there exists $u \in supp^+(x-y)$ and $v \in supp^-(x-y)$ such that $x - \chi_u + \chi_v$ and $y + \chi_u - \chi_v$ are both in B.*
- *For distinct x, y in B, for all $u \in supp^+(x-y)$ there exists $v \in supp^-(x-y)$ such that $x - \chi_u + \chi_v$ is in B.*
- *For distinct x, y in B, for all $u \in supp^+(x-y)$ there exists $v \in supp^-(x-y)$ such that $x - \chi_u + \chi_v$ and $y + \chi_u - \chi_v$ are both in B.*

We should note that in lemma 7 the transition from second to third to the first are somewhat trivial yet the real surprising result is the transition from the first to the second.

By definition $M^\natural$-convex functions are also M-convex. It is also possible to add a slack variable and transform M-convex functions into $M\{\natural\}$-convex functions. Some interesting remarks on M/$M^\natural$ convex functions are as follows.

- Effective domain of M/$M^\natural$ convex functions are M/$M^\natural$ convex sets.
- An M/$M^\natural$ convex function need not stay M/$M^\natural$ convex when its effective domain is restricted to an arbitrary M/$M^\natural$ convex.
- Sum of M/$M^\natural$ convex functions are not necessarily M/$M^\natural$ convex.

Dual of a function f is defined as $f^*(y) = argmax_x(y^t x - f(x))$. For convex functions we already know that $(f^*)^* = f$. Unsurprisingly this result applies to M-convex functions.

Theorem 3 *([40]) For an M-convex function $f : Z^n \rightarrow R$ and $f^*(y) = argmax_{x \in Z^n}(y^t x - f(x))$ we have $(f^*)^* = f$.*

We define the base polyhedron of a submodular system as follows.

Definition 14 *Base polyhedron of a submodular function f is*

$$\{x \in R^{\|E\|} \| x(S) \leq f(S) \forall S \subseteq E, x(E) = f(E)\}$$

We have a strong connection between base polyhedra and M-convex sets given as follows.

Theorem 4 *[40] Every M-convex set can be represented as the integral points in a base polyhedron of an integral submodular system.*

3.3.3 Examples of algorithms for optimizing M/$M^\natural$-convex functions

One of the most important properties of M/$M^\natural$-convex functions is that the local(by local we mean the points that are at most differ by one in two dimensions) optimum is

also a global optimum. This leads to fast , greedy polynomial algorithms for optimizing such functions. Algorithm 8 is a straightforward extension of steepest descent to M-convex functions. By searching the 2-depth neighbors as in the steepest descent we also

Algorithm 8: M-convex Steepest descent [40]

f is M-convex, choose $x_0 \in R^n$
for $k \geq 0$ **do**
  if $f(x) = min_{u,v} f(x - \chi_u + \chi_v)$ **then**
    End and output x_k as minimizer
  end if
  $u, v = argmin_{s,t} f(x_k - \chi_t + \chi_s)$
  $x_{k+1} = x_k - \chi_t + \chi_s$
end for

bound the domain that contain the optimizer which leads to the following algorithm.

Algorithm 9: Domain reduction algorithm for M-convex functions[40]

Set $B_0 = Domf$
for $k \geq 0$ **do**
  if $f(x) = min_{u,v} f(x - \chi_u + \chi_v)$ **then**
    End and output x_k as minimizer
  end if
  $u, v = argmin_{s,t} f(x_k - \chi_t + \chi_s)$
  $B_{k+1} = B_k \cap \{y \in Z^V \| y(u) \leq x(u) - 1, y(v) \geq x(v) + 1\}$
end for

Both of these algorithms are polynomial time on the dimension and the size of the domain.

3.4 M-convex Submodular Flow Problem

Now we shift our attention to networks. We represent a directed graph with the tuple $G = (V, A)$. The flows on the links are represented as the vector x and net-flows on the nodes are represented by y.

3.4.1 Submodular Flow problem

Let B be the base polyhedron of a submodular integral system. The submodular flow problem is then defined as follows.

$$\text{Minimize} \tag{3.1}$$

$$\sum_{a \in A} x(a)c(a) \tag{3.2}$$

$$\text{subject to:} \tag{3.3}$$

$$\underline{\mathrm{k}}(a) \leq x(a) \leq \bar{k}(a), (\forall a \in A) \tag{3.4}$$

$$y(v) = \sum_{i \in \delta^+(v)} x_i - \sum_{i \in \delta^-(v)} x_i, (\forall v \in V) \tag{3.5}$$

$$y \in B \tag{3.6}$$

The lower and upper bounds on the link flows are represented as functions $\underline{\mathrm{k}}$ and $\bar{k}$. In this simplest model the objective is a linear function of link flows, coefficients are given as $c(a)$, and the netflows are coming from a submodular base polyhedron B.

3.4.2 Optimality Conditions of Submodular Flow problem

The optimality of the Submodular flow problem can be told by looking at the residual graph defined as follows.

Definition 15 *For feasible link flows x and corresponding net flows y in G, auxiliary graph for the flow problem where we have independent bounds for link capacities is the weighted directed graph $(V, A^{res} \cup B^{res})$ where*

- $A^{res} = \{a \| x(a) < \bar{k}_a, a \in A\}$

- $B^{res} = \{-a \| x(a) > \underline{k}_a, a \in A\}$

and $\forall e \in A^{res}$ *the link cost is* $c(e)$, $\forall e \in B^{res}$ *the link cost is* $-c(-e)$.

A link flow x and net flow y is optimal if and only if there are no negative cost cycles in the residual graph defined in 15. This follows from the standard optimality condition for networks and the objective being linear.

3.4.3 M-convex Submodular flow problem

As we are going to see, the preferences of agents in our problem can not be represented as a linear function. To address this problem the following M-convex Submodular Flow problem is proposed by (Candogan,Epitropou,Vohra [41]).

$$\text{Minimize} \tag{3.7}$$

$$\sum_{a \in A} x(a)\sigma(a) + f(y) \tag{3.8}$$

$$\text{subject to:} \tag{3.9}$$

$$\underline{c}(a) \leq x(a) \leq \bar{c}(a), (\forall a \in A) \tag{3.10}$$

$$y(v) = \sum_{i \in \delta^+(v)} x_i - \sum_{i \in \delta^-(v)} x_i, (\forall v \in V) \tag{3.11}$$

$$y \in B \tag{3.12}$$

An important note here is that $f : B \to R$ is an M-convex function(and by definition its domain is the base polyhedra B itself).

3.4.4 Optimality Conditions for the M-convex Submodular Flow problem

The optimality conditions for the M-convex flow problem is also connected to negative cycles and in auxiliary graph which is an extension of the residual graph. Although as we are going to see the conditions are more complex here compared to the simpler submodular flow problem.

Definition 16 *For feasible link flows x and corresponding net flows y in G, auxiliary graph for the flow problem where we have independent bounds for link capacities is the weighted directed graph $(V, A^{aux} \cup B^{aux} \cup C^{aux})$ where*

- $A^{aux} = \{a \| x(a) < \bar{k}_a, a \in A\}$
- $B^{aux} = \{-a \| x(a) > \underline{k}_a, a \in A\}$
- $C^{aux} = \{(u, v) \| u, v \in V, f(y - \chi_v + \chi_u) - f(y)\}$

and the corresponding link costs are

$$c_a^{aux}(x, y) = \tag{3.13}$$

$$c(a), \qquad a \in A^{aux} \tag{3.14}$$

$$-c(-a), \qquad a \in B^{aux} \tag{3.15}$$

$$f(y - \chi_v + \chi_u) - f(y), \qquad a \in C^{aux}. \tag{3.16}$$

$$\tag{3.17}$$

Since we have an additional non-linear component in the objective function, in the auxiliary graph we create new link according to the change in the function. For this formulation the optimality conditions are given as follows.

Theorem 5 *(Murota [39]) The following three conditions are equivalent.*

- (x, y) *is an optimal solution to M-convex Submodular Flow Problem.*
- *There is no negative cycle in* $G^{aux}(x, y)$
- *There exists a vector* $p \in R^{\|V\|_1}$ *such that for all* $(u, v) \in A$ *we have*
  1. $c((u, v)) + p_u - p_v > 0 \rightarrow x((u, v)) = \underline{k}_{(u,v)}$
  2. $c((u, v)) + p_u - p_v < 0 \rightarrow x((u, v)) = \bar{k}_{(u,v)}$
  3. *moreover* $f(y) - p^t y \leq f(y') - p^t y'$ *for all* $y' \in Z^V$.

The second condition shows that the connection between the existence of negative cycles and optimality is still valid. The third condition is the result of optimality conditions w.r.t. Lagrangian duality (the vector p is the dial variables for the M-convex submodular flow problem).

3.5 Economics Background

At this point we have gone through most of the optimization background that we are going to make use of in this chapter. Now we turn our attention to Economics side of the problem.

One of the major questions in economics, is what point (a point is a combination of variables such as price, allocation, flows...) the economy will converge and become balanced. At such a point, in the absence of externalities system will stay put. For example Nash equilibrium is very important because at such an equilibrium no agent will have any incentive to deviate from their strategy assuming the rest of the agents stay. It is not possible to review all of the concepts regarding equilibrium in this chapter.

For fundamentals of Game Theory reader should refer to (Osborne,Rubinstein [42]) for a gentle introduction and (Fudenberg,Tirole [43]) for a through treatment.

3.5.1 Competitive Equilibrium

In a trade network where the agents are both sellers and buyers, are in constant competition. The utility of each agent is the sum of its profits for the goods it sells and the utility it receives from the goods it has. For such agents we are going to define competitive equilibrium as follows.

Definition 17 *In a trade network, net flows y corresponding link flows x and the price on link flows p constitute Competitive Equilibrium if the following two conditions are satisfied.*

- *For every agent(node) x, y, p maximizes its utility.*
- *Market clears, that is the amount of goods produced is equal to the goods purchased.*

Since the trade networks we are considering are based on competition another important property that we will be assuming is the goods being gross substitutes.

3.5.2 Gross substitutes property

The demand goods, in this case each link represent the selling and buying of a particular good, are related to prices as follows.

Definition 18 *At a competitive equilibrium, when the price of a particular good increases while the remaining prices stay the same, the demand for that good decreases and the demand for the remaining goods increases or stay the same.*

Here an important remark is that the utility functions of agents are $M^\natural$ concave if and only if the gross substitutes property is satisfied (Yang,Fujishige[35]). This further justifies the validity of the trade network model we are investigating here. One of the benefits of gross substitutes property is that once it is satisfied the prices that lead to Competitive Equilibrium can be computed in polynomial time via "Tattonement process"

Definition 19 *Let $d(p)$ be the demand function (in this case it outputs the link flows x), the Tattonement procedure is defined as follows.*

1. *Start with a small enough price vector which guarantees excess demand for all goods*
2. *Compute the demand $d(p)$*
3. *If there is no excess demand for any of the goods then stop.*
4. *Else pick a good with excess demand and increase its prices by a small amount. Return to step 2.*

Theorem 6 *Let $d(p)$ be the demand function for prices p. If gross substitutes and the following conditions are satisfied then Tattonement procedure achieves Competitive equilibrium.*

- *The demand function is homogeneous, $d(\alpha p) = d(p)$*
- *Walras's law is satisfied , $p^T d(p) = 0$.*

These are optimality conditions for a convex function. The procedure was introduced by Walras for auctions, the auctioneer announced the prices , collected the demands and kept increasing prices until there is no excess demand(prices are increased at small enough increments so demand is equal to the supply), at this last step the trades are executed.

3.5.3 Representing Trade Networks with simple constraints as M-convex Submodular Flows

Here we will demonstrate how trade networks with simple constraints can be represented by an instance of M-convex Submodular Flow Problem by following the footsteps in [41]. In a trade network every node represents an agent while every edge represents a trade. The network is a multi-graph since there may be different contracts between agents.

For every agent i each trade it is involved in is an edge from or towards i. The set of trades agent i is involved in is a characteristic vector y^i such that if a trade $e \in \delta(i)$ occurs, $y^i_e = 1$ for $e \in \sigma_+(i)$ and $y^i_e = -1$ for $e \in \sigma_-(i)$. For trades that can have varying number of goods, for each possible trade we add a parallel link to the corresponding nodes.

The utility function of an agent i, call it w_i, is defined on the possible set of trades that involves i and is M-concave. For our transformation we create two new nodes v^i_e, v^k_e for every possible trade e from agent i to agent k. Now every agent has a set of nodes which corresponds to the trades it is involved in, that is $V^i = \{v^i_e \| i \in N, e \in \delta(i)\}$. A remark on the utility function here is that $w_i(y) = -\infty$ if $y^i_e \notin \{-1, 0, 1\}$ for some trade $e \in \delta(i)$ this lets us take the individual capacity bounds on the links $\bar{k} = \infty$ and $\underline{k} = -\infty$.

3.5.4 Some results regarding Trade Networks with simple constraints

Now that we have established a transformation from Trade networks to MSFP we will go over some immediate consequences which are established in [41].

Theorem 7 *(Thm 4.1 in [41])In 3.1 if there is no bound for link capacities and the*

utility is a separable M-convex function then there exists a competitive equilibrium.

Proof: This result follows from the optimality conditions given in 5. If the link capacities are not bounded then the dual variable p will be the same for both of the end nodes of an edge. The final condition given $f(y)$ being separable shows that the resulting net flows will be optimum for every agent. Together we conclude that the p and x, y constitutes a competitive equilibrium where p are the prices we are looking for. ■

Corollary 3 *In trade networks with simple constraints there exists a Competitive Equilibrium.*

This from the transformation having unbounded link capacities and theorem 7

Definition 20 *And outcome is efficient if it maximizes $\sum_i f(y)$ in the Submodular Flow problem.*

The following theorem shows that competitive equilibrium is also efficient.

Theorem 8 *(thm 4.2 in [41])Under the conditions in theorem 7 the Competitive Equilibrium is efficient.*

The proof again follows from the last optimality condition of 5. At this point we should remind the reader that we are interested in the goods/services which are not divisible. Mathematically this means that we are interested in only integral Competitive Equilibrium. Which leads us to the following lemma.

Lemma 8 *If the bounds on the link capacities are integer valued then there exist an integral optimum for the Submodular Flow Problem.*

The proof follows from the existence of a negative cycles if the solution is not optimum. If one starts with an integral x then if there is a negative cycle, because the link capacities

are bounded by integers then one can always traverse that negative cycle by a unit flow. This means that by traversing these cycles by one unit at a time, we can reach the optimum flow moreover after each traversal the flow will stay integral and the result follows. We finish this section by stating the following corollary.

Corollary 4 *Competitive Equilibrium for trade networks with simple constraints is efficient.*

3.6 Networks with complex constraints

So far we have seen that when we have only upper and lower bounds on individual link capacities then for the trade networks where utilities are separable M-convex functions, there exists a competitive equilibrium moreover it can be computed in polynomial time. The question that we want to address in this chapter is what happens when we have complex constraints on the link capacities. Suppose the node represents a farmer and there are two different produces he/she wants to sell say apples and oranges. The farmer is capable of producing 10 tons of apples or 10 tons of oranges each year however it is not possible to produce 10 tons both at the same time because of the land/labor etc. limitations. If the farmer is asked to produce both suppose he/she is capable of producing a total of 15 tons of apples and oranges. The trade network which boils down to the submodular flow problem can not be used as is for this situation. It is not hard to come up with examples like this so it is left to the reader to find other examples.
Unfortunately we can not have competitive equilibrium for arbitrary constraints so we should restrict our attention to a subset of constraints.

3.6.1 Polymatroidal Constraints

Now that we have built our foundations we can start presenting our contributions. We are going to be looking into constraints which restrict the link flows to a polymatroid. The farmer we have talked about earlier was having an additional constraint when he/she wanted to produce both apples and oranges. The total amount of apples and oranges was less than the sum of the maximum amount of oranges and apples he/she can produce individually. This example motivates the use of polymatroidal constraints on the trades(links in the network).

The concept of polymatroidal constraints are first introduced by Lawler ([44]). This work investigates network flow on a single source and single sink network and establishes that optimality of a flow on such a network still boils down to not having an augmenting path. Moreover for integral polymatroidal constraints the optimal flow will also be integral. One important note is that in this case the concept of auxiliary graph is nonexistent. Moving on a link under polymatroidal constraints may change the feasible links afterwards unlike the case of individual capacity bounds on the links. Another important note here is that in Lawler's model for each node, the incoming and outgoing links are coming from separate polymatroids,in other words there are no constraints bounding a collection of both incoming and outgoing links.

3.6.2 A counterexample for arbitrary Polymatroidal Constraints

The first question we asked was if it was possible to have arbitrary polymatroidal constraints on the links without taking their orientation into account. That is for the network with integral polymatroidal constraints can we guarantee to have an integer efficient flow. As we show in the following counterexample the answer to this question is negative.

Take the simple network given in figure 3.4 where there are 2 agents, N_1, N_2, and each agent has one incoming and one outgoing link. The $M^\natural$-convex utility functions of the agents are defined as w_i as follows.

$$w_1(y_1, y_2) = \tag{3.18}$$

$$0.5 \qquad , y_1 = 0, y_2 = 0 \tag{3.19}$$

$$0 \qquad y_1 = 1, y_2 = -1 \tag{3.20}$$

$$1 \qquad y_1 = 1, y_2 = 0 \tag{3.21}$$

$$1 \qquad y_1 = 0, y_2 = -1 \tag{3.22}$$

$$\infty \qquad elsewhere. \tag{3.23}$$

$$w_2(y_3, y_4) = \tag{3.24}$$

$$0.5 \qquad , y_3 = 0, y_4 = 0 \tag{3.25}$$

$$0 \qquad y_4 = 1, y_3 = -1 \tag{3.26}$$

$$1 \qquad y_4 = 1, y_3 = 0 \tag{3.27}$$

$$1 \qquad y_4 = 0, y_3 = -1 \tag{3.28}$$

$$\infty \qquad elsewhere. \tag{3.29}$$

Since w_1 and w_2 are defined on orthogonal domains then their sum $f(y) = w_1(y_1, y_2) + w_2(y_3, y_4)$ is also $M^\natural$-Convex. The link flows are bounded by the polymatroid

$$P = \{x \| x_1 \leq 1, x_2 \leq 1, x_1 + x_2 \leq 1\}.$$

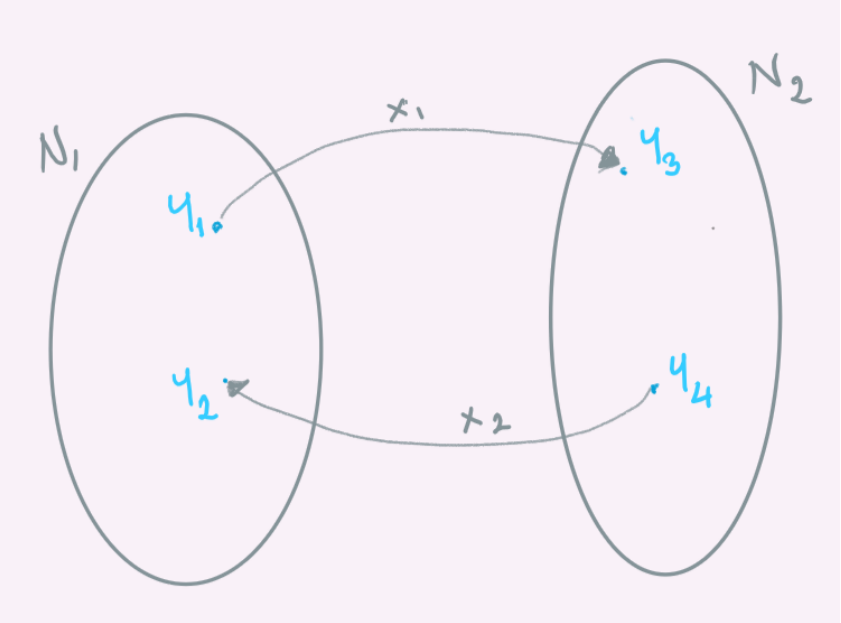


Figure 3.4: Simple Counterexample

Now with slight abuse of notation let $y(x)$ be the net flows on y when we have link flows x. Then we have $f(y(0,0)) = 1, f(y(1,1)) = 0, f(y(1,0)) = f(y(0,1)) = 2$. However $min_{x \in P} \hat{f}(y(x)) \leq 1/2 f(y(0,0)) + 1/2 f(y(1,1)) = 0.5$ and this is smaller than each integral feasible flow will lead to a larger objective value. This shows that in this example we can not have an integral efficient outcome.

This counterexample shows us that we should be focusing our attention to less restricted cases.

3.7 When set of Net-flows is a polymatroid

Although we have seen that having general polymatroidal constraints on the link flows may cause a Submodular Flow problem not to have an integral solution, we are going to prove that if the set of feasible net-flows is a polymatroid then we do have an integral optimum. This instance of the submodular flow problem can be simplified to the

following in this case.

$$\text{minimize } \tilde{f}(y) \tag{3.30}$$

$$st, \qquad y \in B \tag{3.31}$$

where B is an integral polymatroid. Let's start with a theorem we are going to make use of in this subsection.

Theorem 9 *(Murota,Thm 4.22, [37]) For M-convex sets B_1 and B_2 the convex closure of their intersection is equal to the intersection of their convex closures, that is*

$$\bar{B_1} \cap \bar{B_2} = \overline{(B_1 \cup B_2)}.$$

Now let

$$B_f(x) = \{z \in N(x) \| \exists \lambda, \sum_z \lambda_z = 1, \sum_z \lambda_z z = x, \sum_z \lambda_z f(z) = \tilde{f(x)}\}$$

where $N(x)$ is the unit cube surrounding point x and as we have defined earlier $\tilde{f}(x)$ is the convex envelope of function f.

Lemma 9 *For every M-convex function f and x in the convex hull of the domain f, $B_f(x)$ is an M-convex set.*

Proof: Let $x_1, x_2 \in B_f(x)$. Since $f(x)$ is an M-convex function there exists $u \in supp_+(x_1 - x_2)$ and $v \in supp_-(x_1 - x_2)$ such that,

$$f(x_1 - \chi_u + \chi_v) + f(x_2 + \chi_u - \chi_v) \leq f(x_1) + f(x_2).$$

Now take convex coefficients λ^1, λ^2 such that $\sum_{z\in N(x)} \lambda^1_z z = \sum_{z\in N(x)} \lambda^2_z z = x$ and $\sum_{z\in N(x)} \lambda^i f(z) = \tilde{f}(x)$ (by definition we have such λ^i). Then for $\lambda = \frac{\lambda^1+\lambda^2}{2}$ we have $\sum_z \lambda_z z = x$, $\sum_{z\in N(x)} f(z) = \tilde{f}(x)$, $\lambda_{x_1} > 0$ and $\lambda_{x_2} > 0$.
By taking $\lambda_\delta = min\{\lambda_{x_1}, \lambda_{x_2}\}$ we get the following inequality

$$\tilde{f}(x) \leq (\sum_{z\in N(x)-\{x_1,x_2\}} \lambda_z f(z)) + \lambda_\delta(f(x_1-\chi_u+\chi_v)+f(x_2+\chi_u-\chi_v)) + (\lambda_{x_1}-\lambda_\delta)f(x_1) + (\lambda_{x_2}-\lambda_\delta)f(x_2)$$

$$\leq \sum_{z\in N(x)} f(z) = \tilde{f}(x).$$

But then, by definition $(x_1 - \chi_u + \chi_v), (x_2 + \chi_u - \chi_v)$ are both in $B_f(x)$ hence it$(B_f(x))$ has to be an M-convex set. ■

Corollary 5 *For every convex coefficient λ such that $\sum_{z\in B_f(x)} \lambda_z z = x$ we have $\tilde{f}(x) = \sum_z \lambda_z f(z)$.*

Proof follows trivially from the definition of $B_f(x)$.

Theorem 10 *If f is and M-convex function and B is an integral polymatroid then there exists an integer optimizer for problem 3.30.*

Proof: Let x be an optimizer for problem 3.30. If $N(x) \subseteq B$ then we are done since at least one extreme point of $N(x)$ will also be an optimizer. Else let F_x be the face of B that contains x. F_x is an M-convex set (as we have seen earlier M-convex sets are essentially polymatroids).
From lemma 9 we know that $B_f(x)$ is an M-convex set and from theorem 9 we have

$$x \in \overline{F_x} \cap \overline{B_f(x)} = \overline{(F_x \cup B_f(x))}.$$

But then there exists convex coefficients λ such that

$$\sum_{z\in(F_x\cup B_f(x))} \lambda_z z = x.$$

From corollary 5 we have

$$\sum_{z\in(F_x\cup B_f(x))} \lambda_z f(z) = \tilde{f}(x).$$

But then there exists a $y^* \in (F_x \cup B_f(x))$ such that $f(y^*) \leq \tilde{f}(x)$ and since x is an optimizer, y^* has to be an optimizer too. Since y^* is integral this concludes the proof. ■

This result shows that we may have more complicated constraints on the links and still have an integral optimum hence may have an integral CE. On the other hand, the constraints on link flows that will lead to the set of feasible net-flows being a polymatroid is not clear. We will discuss this later.

3.8 When the set of incoming(or outgoing) links are a polymatroid

Although link flows can not have arbitrary polymatroidal constraints, the case where only outgoing (or incoming) link flows are from a polymatroid is a good candidate for study.

Let $g_i(S) \geq \sum_{j\in S\cap\delta_i^+} x(j)$ be the submodular and integral function that bounds node i's outgoing links. Since $g_i(.)$ is defined on the outgoing links of i, it is indifferent towards the flows on the other links, that is $g_i(S) = g_i(S \cap \delta_i^+)$. As before we represent the net-flows on the node i as $y(i) = \sum_{j\in\delta_i^+} x(j) - \sum_{j\in\delta_i^-} x(j)$.

Lemma 10 *If there are two nodes in the graph and for each node i set of feasible outgoing edge flows is an integral polymatroid defined by a submodular and integer val-*

ued function $g_i(.)$ then the following holds true. Let two feasible flows $y, \hat{y}$ be such that $y(1) > \hat{y}(1)$ ($y(2) < \hat{y(2)}$). There exists an edge j such that one of the following holds true,

- *j is from 1 to 2, $x(j) > \hat{x}(j)$ and $\{x - e_j, \hat{x} + e_j\}$ are feasible link flows*
- *j is from 2 to 1, $x(j) < \hat{x}(j)$ and $\{x + e_j, \hat{x} - e_j\}$ are feasible link flows.*

Proof: Observe that for the given net flows we have whether $\sum_{i\in\delta_1^+} x(i) > \sum_{i\in\delta_1^+} \hat{x}(i)$ or $\sum_{i\in\delta_2^+} x(i) < \sum_{i\in\delta_2^+} \hat{x}(i)$. To see the existence of such a link j, let's focus on the first case where we have , $\sum_{j\in\delta_1^+} x(j) > \sum_{j\in\delta_1^+} \hat{x}(j)$ since the other case is symmetric to this one. If x is an interior point of the feasible polymatroid then we are done since we can use any j such that $x(j) > \hat{x}(j)$ and the exchange will be feasible. If x is on a facet of the polymatroid let $X_+ = \{j : x(j) > \hat{x}(j)\}$ and assume that for any j in X_+ , $\hat{x} + e_j$ is not feasible(otherwise we are done). This means that for all j in X_+ we have an $S_j \subseteq E$ such that

$$\hat{x}(S_j) = g_1(S_j)$$

and because $g_i(.)$ is submodular we further have

$$\hat{x}(S) = g_1(S)$$

for $S = \cup_j S_j$. But then

$$\sum_{j\in E} \hat{x}(j) = \sum_{j\in E\setminus S} \hat{x}(j) + \sum_{j\in S} \hat{x}(j) = \sum_{j\in E\setminus S} \hat{x}(j) + g(S) \geq \sum_{j\in E\setminus S} x(j) + \sum_{j\in S} x(j) = \sum_{j\in E} x(j)$$

which is a contradiction.

■

Corollary 6 *10 holds true when the incoming links are bounded by a submodular function.*

The case where incoming links are bounded is symmetric to the original case hence same argument carries on trivially.

Corollary 7 *If there are two nodes in the graph and set of feasible edge flows is an integral polymatroid defined by submodular and integer valued functions $g_i(.)$ then the set of feasible, integral net flows is an M-convex set.*

Proof: Let's have two net flows y and $\hat{y}$. If $y = \hat{y}$ we are done. Else w.l.o.g. assume that $y(1) > \hat{y}(1)$ $(y(2) < \hat{y}(2))$. Since g_i is supermodular then whether there is an edge $j \in \delta_1^+$ such that $x(j) > \hat{x}(j)$ and $x - e_j, \hat{x} + e_j$ are feasible or there is an edge $j \in \delta_2^+$ such that $x(j) < \hat{x}(j)$ and $\hat{x} + e_j, x - e_j$ is feasible. Then by increasing x_j and decreasing $\hat{x}_j$ by one, and vice versa for the opposing case, we get the feasible net flows $(y(1) - 1, y(2) + 1)$ and $(\hat{y}(1) + 1, \hat{y}(2) - 1)$.

■

This theorem and its corollaries essentially prove that for two nodes, polymatroidal constraints on the incoming/outgoing link flows will lead to the set of feasible net-flows being a polymatroid hence our result in (10) follows for such constraints.
The next is the case where we have multiple nodes and two net-flows. We show that when we partition the nodes into two sets where the cumulative net-flows are different for each set, we still have the exchange property.

Lemma 11 *Let set of feasible edge flows be an integral polymatroid defined by submodular and integer valued functions $g_i(.)$ on outgoing links as given earlier. Let $\{V^+, V^-\}$ be*

a partition of V *(*$V^+ \cup V^- = V$ *and* $V^- \cap V^+ = \emptyset$ *),* y *and* $\hat{y}$ *be two net flows(*$x, \hat{x}$ *corresponding edge flows) such that* $\sum_{i \in V^+} y(i) > \sum_{i \in V^+} \hat{y}(i)$ *and* $\sum_{i \in V^-} y(i) < \sum_{i \in V^-} \hat{y}(i)$*. Then one of the following is satisfied.*

- *There exists* $s \in V^+$*,* $t \in V^-$ *such that whether* $\{x - e_j, \hat{x} + e_j\}$ *are feasible for* $j = (s,t)$ *or there exists a node* $a \in V^+$ *and links* $j = (a,t)$*,*$j' = (a,s)$ *such that* $\{x - e_j + e_{j'}, \hat{x} + e_j - e_{j'}\}$ *are feasible.*

- *There exists* $s \in V^-$*,* $t \in V^+$ *such that whether* $\{x - e_j, \hat{x} + e_j\}$ *are feasible for* $j = (s,t)$ *or there exists a node* $a \in V^-$ *and links* $j = (a,t)$*,*$j' = (a,s)$ *such that* $\{x - e_j + e_{j'}, \hat{x} + e_j - e_{j'}\}$ *are feasible.*

Proof: Create two supernodes, y_+, y_- by combining the nodes in set V^+ and V^-(by combining we mean deleting the interlinks within V^+ and V^- and connecting any link between them to the created supernodes). For the surviving links define the submodular function $\hat{g}_+(X) = \sum_{i \in (V^+ \times V^-)} g_i(X \cap \delta_i^+)$ and $\hat{g}_-(X) = \sum_{i \in (V^- \times V^+)} g_i(X \cap \delta_i^+)$. By lemma(10) we know that there exists a j such that $j \in (V^+ \times V^-)$, $x(j) > \hat{x}(j)$, $x - e_j, \hat{x} + e_j$ are feasible under $\hat{g}(.)$ or vice versa. If $x - e_j, \hat{x} + e_j$ $(x + e_j, \hat{x} - e_j)$ are also feasible under $g(.)$ then we are done. Else, wlog consider the first case, because polymatroids being $M - convex$ we know that there exists a j' such that $x - e_j + e_{j'}, \hat{x} + e_j - e_{j'}$ are feasible. Again if j' lies in V^+ then we are done.

Now let's show that there always exists such an edge j' in V^+. If $\hat{x} + e_j$ is not feasible then take the minimal saturated set, B, (Lawler's terminology for the minimal set B such that $\hat{x}(B) = g_i(B)$ where i is the head of link j). B can not be a subset of $(V^+ \times V^-)$ otherwise we have $g_i(B) = \hat{x}(B) < \hat{g}_i(B) = g_i(B)$. Then for any j' in B $(V^+ \times V^-)$ we have $\hat{x} + e_j - e_{j'}$ feasible.

■

3.9 An immediate result

Let's first give a similar representation of trade networks having polymatroidal constraints on the outgoing edges.

For every agent i and every incoming edge (trades where i is the buyer) $e \in \delta_-(i)$create a new node y_e^i. Create a node y_o^i and for all of the outgoing edges from i(the trades where agent i is the seller), $(i,j) = e \in \delta_+(i)$ create an edge between y_o^i and y_e^j. What we are doing here is to multiply the nodes for the buyers for each trade and keeping only one node for each seller. Each agent i is now represented by a set of nodes $\{y_o^i\} \cup_{e\in\delta_-(i)} \{y_e^i\}$ and its utility function is $w_i(y^i)$ which is indifferent towards net-flows of the nodes of other agents.

In this formulation, for every i the set $S^i = \{y_o^i\} \cup_{(i,j)\in\delta_+(i)} \{y_e^j\}$ is a connected component. If we take two different net-flows, $y, \hat{y}$, for each z such that $y(z) > \hat{y}(z)$ there exists an i such that $z \in S^i$. From lemma 11 we know that there exists a $z' \in S^i$ whether $x - e_{(z,z')}, \hat{x} + e_{(z,z')}$ or $x - e_{(a,z')} + e_{(z,a)}, \hat{x} + e_{(a,z')} - e_{(z,a)}$ are feasible and in both cases we have a simple exchange between net-flows on z and $\hat{z}$. This shows that for this case the set of net-flows are essentially M-convex.

From theorem 10 we know that for this case we have an integral optima and from the duality conditions this optima can be supported via a price vector p which shows that we have competitive equilibrium.

3.9.1 Discussion and Future work

In this chapter we have laid down some of the results from our ongoing research on trade networks with complex constraints. We have shown that for the M-convex Submodular Flow problem if the set of net-flows is M-convex then we always have an integral

optima. This result by itself is an important one since it shows that we can solve covex-extended version of the problem via continuous methods to reach an integral solution. Moreover our lemma 11 shows that for some networks where we have polymatroidal constraints on the outgoing link constraints this still leads to set of net-flows being M-convex. As we have seen in the previous section such a result can be used to show the existence of Competitive equilibrium.

This work is far from over though. The main objective is to come up with a general existence result for polymatroidal constraints which we believe is an interesting question and probably has a positive answer to it.

Bibliography

[1] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods.*, *USSR Computational Mathematics and Mathematical Physics* **5** (1964), no. 4 1–17.

[2] Y. Nesterov, *A method of solving a convex programming problem with convergence rate o (1/k2)*, in *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.

[3] W. Su, S. Boyd, and E. J. Candes, *A differential equation for modeling nesterov's accelerated gradient method: theory and insights*, *Journal of Machine Learning Research* **17** (2016), no. 153 1–43.

[4] P. Giselsson and S. Boyd, *Monotonicity and restart in fast gradient methods*, in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 5058–5063, IEEE, 2014.

[5] B. O'donoghue and E. Candes, *Adaptive restart for accelerated gradient schemes*, *Foundations of Computational Mathematics* **15** (2015), no. 3 715–732.

[6] A. C. Wilson, B. Recht, and M. I. Jordan, *A lyapunov analysis of momentum methods in optimization*, *arXiv preprint arXiv:1611.02635* (2016).

[7] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, *Mathematical Programming* **146** (2014), no. 1-2 459–494.

[8] K. Hou, Z. Zhou, A. M.-C. So, and Z.-Q. Luo, *On the linear convergence of the proximal gradient method for trace norm regularization*, in *Advances in Neural Information Processing Systems*, pp. 710–718, 2013.

[9] Z. Zhou, Q. Zhang, and A. M.-C. So, *1,p-norm regularization: Error bounds and convergence rate analysis of first-order methods.*, in *ICML*, pp. 1501–1510, 2015.

[10] N. Arkadi and A. Ben-Tal, *Lectures on modern convex optimization*, .

[11] Y. Nesterov, *Smooth minimization of non-smooth functions*, *Mathematical Programming* **103** (2005), no. 1 127–152.

[12] P. J. Huber, *Robust statistics*. Springer, 2011.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, *SIAM review* **43** (2001), no. 1 129–159.

[14] K. Bryan and T. Leise, *Making do with less: An introduction to compressed sensing*, *Siam Review* **55** (2013), no. 3 547–566.

[15] E. Candes and T. Tao, *Decoding by linear programming*, *arXiv preprint math/0502327* (2005).

[16] D. L. Donoho *et. al.*, *Compressed sensing*, *IEEE Transactions on information theory* **52** (2006), no. 4 1289–1306.

[17] D. L. Donoho, *For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution*, *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59** (2006), no. 6 797–829.

[18] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[19] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*, *Bull. Am. Math* **54** (2017) 151–165.

[20] D. Singaraju, R. Tron, E. Elhamifar, A. Y. Yang, and S. S. Sastry, *On the lagrangian biduality of sparsity minimization problems*, in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3429–3432, IEEE, 2012.

[21] S. Chretien, *An alternating l_1 approach to the compressed sensing problem*, *IEEE Signal Processing Letters* **17** (2009), no. 2 181–184.

[22] J.-J. Fuchs, *On sparse representations in arbitrary redundant bases*, *IEEE transactions on Information theory* **50** (2004), no. 6 1341–1344.

[23] B. Chen, K. Madsen, and S. Zhang, *On the characterization of quadratic splines*, *Journal of optimization theory and applications* **124** (2005), no. 1 93–111.

[24] M. Ç. Pinar, *Necessary and sufficient conditions for noiseless sparse recovery via convex quadratic splines*, *SIAM Journal on Matrix Analysis and Applications* **40** (2019), no. 1 194–209.

[25] K. Madsen and H. B. Nielsen, *A finite smoothing algorithm for linear l_1 estimation*, *SIAM Journal on Optimization* **3** (1993), no. 2 223–235.

[26] K. Madsen, H. B. Nielsen, and M. Ç. Pinar, *New characterizations of l1 solutions to overdetermined systems of linear equations*, *Operations Research Letters* **16** (1994), no. 3 159–166.

[27] K. Madsen, H. B. Nielsen, and M. Ç. Pinar, *A finite continuation algorithm for bound constrained quadratic programming*, *SIAM Journal on Optimization* **9** (1998), no. 1 62–83.

[28] I. Selesnick, *Sparse regularization via convex analysis*, *IEEE Transactions on Signal Processing* **65** (2017), no. 17 4481–4494.

[29] O. L. Mangasarian and R. Meyer, *Nonlinear perturbation of linear programs*, *SIAM Journal on Control and Optimization* **17** (1979), no. 6 745–752.

[30] W. Li and J. Swetits, *The linear l 1 estimator and the huber m-estimator*, *SIAM Journal on Optimization* **8** (1998), no. 2 457–475.

[31] D. Bertsimas and J. Tsitsiklis, *Introduction to linear programming*, *Athena Scientific* **1** (1997) 997.

[32] A. Williams, *Complementarity theorems for linear programming*, *SIAM Review* **12** (1970), no. 1 135–137.

[33] J. W. Hatfield, S. D. Kominers, A. Nichifor, M. Ostrovsky, and A. Westkamp, *Stability and competitive equilibrium in trading networks*, *Journal of Political Economy* **121** (2013), no. 5 966–1005.

[34] A. S. Kelso Jr and V. P. Crawford, *Job matching, coalition formation, and gross substitutes*, *Econometrica: Journal of the Econometric Society* (1982) 1483–1504.

[35] S. Fujishige and Z. Yang, *A note on kelso and crawford's gross substitutes condition*, *Mathematics of Operations Research* **28** (2003), no. 3 463–469.

[36] K. Murota and A. Tamura, *New characterizations of m-convex functions and their applications to economic equilibrium models with indivisibilities*, *Discrete Applied Mathematics* **131** (2003), no. 2 495–512.

[37] K. Murota, *Discrete convex analysis*. SIAM, 2003.

[38] S. Moriguchi, K. Murota, A. Tamura, and F. Tardella, *Scaling, proximity, and optimization of integrally convex functions*, *Mathematical Programming* **175** (2019), no. 1-2 119–154.

[39] K. Murota, *Submodular flow problem with a nonseparable cost function*, *Combinatorica* **19** (1999), no. 1 87–109.

[40] K. Murota, *Recent developments in discrete convex analysis*, in *Research Trends in Combinatorial Optimization*, pp. 219–260. Springer, 2009.

[41] O. Candogan, M. Epitropou, and R. V. Vohra, *Competitive equilibrium and trading networks: A network flow approach*, in *Proceedings of the 2016 ACM Conference on Economics and Computation*, pp. 701–702, ACM, 2016.

[42] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT press, 1994.

[43] D. Fudenberg and J. Tirole, *Game theory, 1991*, *Cambridge, Massachusetts* **393** (1991), no. 12 80.

[44] E. L. Lawler and C. U. Martel, *Computing maximal "polymatroidal" network flows*, *Mathematics of Operations Research* **7** (1982), no. 3 334–347.