De-singularity Subgradient for the q-th-Powered ℓ_p-Norm Weber Location Problem

Zhao-Rong Lai[1,2], Xiaotian Wu[1], Liangda Fang[3,4], Ziliang Chen[5], Cheng Li[2*]

[1]College of Cyber Security, Jinan University
[2]Department of Mathematics, College of Information Science and Technology, Jinan University
[3]Department of Computer Science, College of Information Science and Technology, Jinan University
[4]Pazhou Laboratory
[5]Research Institute of Multiple Agents and Embodied Intelligence, Peng Cheng Laboratory
{laizhr, wxiaotian, fangld}@jnu.edu.cn, c.ziliang@yahoo.com, licheng@stu2020.jnu.edu.cn

Abstract

The Weber location problem is widely used in several artificial intelligence scenarios. However, the gradient of the objective does not exist at a considerable set of singular points. Recently, a de-singularity subgradient method has been proposed to fix this problem, but it can only handle the q-th-powered ℓ_2-norm case ($1 \leqslant q < 2$), which has only finite singular points. In this paper, we further establish the de-singularity subgradient for the q-th-powered ℓ_p-norm case with $1 \leqslant q \leqslant p$ and $1 \leqslant p < 2$, which includes all the rest unsolved situations in this problem. This is a challenging task because the singular set is a continuum. The geometry of the objective function is also complicated so that the characterizations of the subgradients, minimum and descent direction are very difficult. We develop a q-th-powered ℓ_p-norm Weiszfeld Algorithm without Singularity (qPpNWAWS) for this problem, which ensures convergence and the descent property of the objective function. Extensive experiments on six real-world data sets demonstrate that qPpNWAWS successfully solves the singularity problem and achieves a linear computational convergence rate in practical scenarios.

1 Introduction

The Weber location problem is a fundamental problem that is extensively investigated in artificial intelligence (Lai et al. 2024), machine learning (Li, Sahoo, and Hoi 2016; Lai et al. 2018c, 2020), financial engineering (Lai and Yang 2023), computer vision (Aftab, Hartley, and Trumpf 2015), and operations research (Ostresh 1978). For a general definition, it seeks a point $\boldsymbol{x}_*$ that minimizes the weighted sum of the q-th power of the ℓ_p distances to m fixed data points $\{\boldsymbol{x}_i\}_{i=1}^m \subseteq \mathbb{R}^d$ (Weber 1909; Morris 1981; Chen 1984; Brimberg and Love 1993), defined as the q-th-powered ℓ_p-norm Weber location problem (**qPpNWLP**):

$$\boldsymbol{x}_* \in \arg\min_{\boldsymbol{y}\in\mathbb{R}^d} C_{p,q}(\boldsymbol{y}) := \sum_{i=1}^m \xi_i \|\boldsymbol{y}-\boldsymbol{x}_i\|_p^q, \qquad (1)$$

where ξ_i denotes the weight for the i-th data point, $\|\cdot\|_p$ denotes the ℓ_p norm, $C_{p,q}(\cdot)$ denotes the cost function related to the q-th power of the ℓ_p norm, $1 \leqslant q \leqslant p$ and $p \geqslant 1$. Due to the same reasons as (Lai et al. 2024), we can assume that the data points $\{\boldsymbol{x}_i\}_{i=1}^m$ are distinct and non-collinear in the rest of the paper.

1.1 The Singularity Problem

There is no closed-form solution to (1), and the gradient-type method is an intuitive and tractable approach. This approach includes those do not explicitly show a gradient form, like the Weiszfeld algorithm (Brimberg and Love 1993). To take a glimpse of the singularity problem, we first compute the gradient of the objective function $C_{p,q}$ as

$$\begin{aligned}&\big(\nabla C_{p,q}(\boldsymbol{y})\big)^{(t)}\\ :=&\sum_{i=1}^m q\xi_i\|\boldsymbol{y}-\boldsymbol{x}_i\|_p^{q-p}|y^{(t)}-x_i^{(t)}|^{p-2}(y^{(t)}-x_i^{(t)}),\end{aligned} \qquad (2)$$

where $\big(\nabla C_{p,q}(\boldsymbol{y})\big)^{(t)}$ denotes the t-th dimension ($1 \leqslant t \leqslant d$) of the gradient. Note that if $q < p$ or $p < 2$, the singularity problem can occur if $\boldsymbol{y}$ hits the following singular set:

$$\begin{cases}\mathcal{S}_p:=\{\boldsymbol{y}\in\mathbb{R}^d|\exists i\in\{1,\ldots,m\}, t\in\{1,\ldots,d\} \text{ s.t. } y^{(t)}=x_i^{(t)}\}\\ \qquad\qquad\qquad\qquad \text{if } 1\leqslant p<2,\\ \mathcal{S}_2:=\{\boldsymbol{x}_i\}_{i=1}^m \qquad\qquad \text{if } 1\leqslant q<2, p=2.\end{cases} \qquad (3)$$

This singularity problem occurs frequently and unexpectedly. Chandrasekaran and Tamir (1989); Vardi and Zhang (2000) indicate that the "bad" points that can yield such singularity in a gradient-type method may constitute a continuum set that can be dense in an open region of $\mathbb{R}^d$ ($d \geqslant 2$) or even the entire $\mathbb{R}^d$. Moreover, this problem cannot be circumvented by straightforward treatments like perturbations and random restarts. More details can be found in (Lai et al. 2024).

If $q \geqslant p$ or $p \geqslant 2$, the term $\|\boldsymbol{y}-\boldsymbol{x}_i\|_p^{q-p}$ or $|y^{(t)}-x_i^{(t)}|^{p-2}$ in (2) will not be singular. For $p \geqslant 2$ but $1 \leqslant q < p$, the singularity problem can be solved by (Lai et al. 2024). Hence this paper mainly focuses on the rest unsolved singular cases with the two key hyperparameters $1 \leqslant q \leqslant p$ and $1 \leqslant p < 2$. Users can select any values of p and q within this range to suit their specific needs. As p approaches 2, the distance between data points becomes closer to the Euclidean distance.

*Corresponding author.

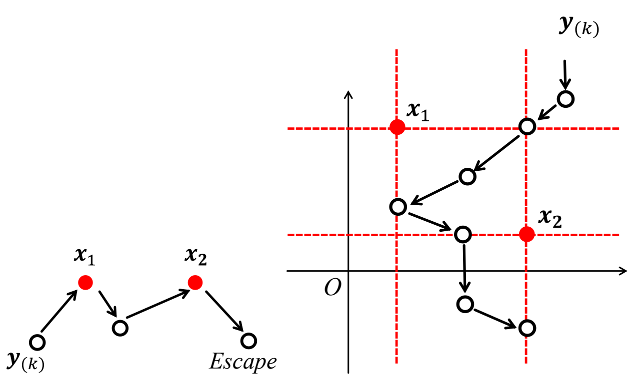
**The supplementary material and codes for this paper are available at https://github.com/laizhr/qPpNWAWS.

Conversely, as p approaches 1, the distance between data points becomes closer to the Manhattan distance, representing a typical non-Euclidean geometry. On the other hand, as q approaches p, the distance between data points gets higher power, which can be advantageous in certain computer vision tasks (Aftab, Hartley, and Trumpf 2015). In summary, allowing for a range of values for p and q enhances the geometrical representation of the Weber location problem.

1.2 Continuum Singular Set for $1 \leqslant p < 2$

(3) indicates that when $p = 2$, the singularity only occurs at the m fixed data points $\{\boldsymbol{x}_i\}_{i=1}^m$. This case has already been completely solved by (Vardi and Zhang 2000; Lai et al. 2024). However, **when $1 \leqslant p < 2$, the singularity occurs in at most $m \cdot d$ hyperplanes that encompass a continuum set of points, which is much more difficult than the $p = 2$ case and no solutions have been developed (see Figure 1).** The difficulties lie in the following three aspects:

1. An effective gradient-type algorithm may visit each singular point for only once. But since there are infinite singular points when $1 \leqslant p < 2$, the algorithm may visit the singular set for infinite times.
2. Based on the first point, there is no unified step size for the escape from the singular set, which affects the convergence property.
3. The geometry of the objective function is complicated when $1 \leqslant q \leqslant p$ and $1 \leqslant p < 2$ so that the characterizations of the subgradients, minimum and descent direction are very difficult, which also affects the convergence property.



(a) $p = 2$ and $1 \leqslant q < 2$ (b) $1 \leqslant p < 2$

Figure 1: (a) When $p = 2$, the singular set $\mathcal{S}_2$ (the red dots) is finite and an effective gradient-type algorithm visits each singular point for only once. Hence the iterate $\boldsymbol{y}_{(k)}$ (the circle) can finally escape from all the singular points. (b) When $1 \leqslant p < 2$, the singular set $\mathcal{S}_p$ is a continuum (the red dashed lines), hence the iterate $\boldsymbol{y}_{(k)}$ may revisit $\mathcal{S}_p$ for infinite times and may not escape from $\mathcal{S}_p$.

To address the above difficulties, we develop a complete de-singularity subgradient methodology for qPpNWLP with $1 \leqslant q \leqslant p$ and $1 \leqslant p < 2$, including all the rest unsolved situations in this problem. **Our main contributions are summarized as follows.**

1. We develop a de-singularity subgradient of the cost function $C_{p,q}$ on the singular set $\mathcal{S}_p$. It can replace the ordinary gradient without increasing computational complexity.
2. We develop a q-th-Powered ℓ_p-Norm Weiszfeld Algorithm without Singularity (q**P**p**NWAWS**). It can identify whether the current iterate is a minimum point; If not, it can further reduce the cost function, no matter whether this iterate is singular or nonsingular. By this way, qPpNWAWS solves the singularity problem.
3. We develop a complete proof for the convergence of qPpNWAWS.
4. We demonstrate that qPpNWAWS achieves a linear computational convergence rate in practical scenarios.

2 Related Works

We review some closely related works on qPpNWLP in this section.

2.1 The ℓ_p-Norm Weiszfeld Algorithm

The ℓ_p-norm Weiszfeld algorithm (pNWA) can be derived by the first-order optimal condition of the qPpNWLP with $1 \leqslant p \leqslant 2$ and $q = 1$ (Brimberg and Love 1993). For a **non-singular optimal point $\boldsymbol{x}_* \notin \mathcal{S}_p$**, setting $\left(\nabla C_{p,1}(\boldsymbol{x}_*)\right)^{(t)} = 0$ in (2) yields

$$x_*^{(t)} = \frac{\sum_{i=1}^m \xi_i \|\boldsymbol{x}_* - \boldsymbol{x}_i\|_p^{1-p} |x_*^{(t)} - x_i^{(t)}|^{p-2} x_i^{(t)}}{\sum_{i=1}^m \xi_i \|\boldsymbol{x}_* - \boldsymbol{x}_i\|_p^{1-p} |x_*^{(t)} - x_i^{(t)}|^{p-2}}, \forall 1 \leqslant t \leqslant d. \tag{4}$$

These are fixed-point equations, which can be converted to a fixed-point algorithm with the following operator $\mathbf{T}_{p,1}$:

$$\begin{aligned} y_{(k+1)}^{(t)} &:= \left(\mathbf{T}_{p,1}(\boldsymbol{y}_{(k)})\right)^{(t)} \\ &:= \frac{\sum_{i=1}^m \xi_i \|\boldsymbol{y}_{(k)} - \boldsymbol{x}_i\|_p^{1-p} |y_{(k)}^{(t)} - x_i^{(t)}|^{p-2} x_i^{(t)}}{\sum_{i=1}^m \xi_i \|\boldsymbol{y}_{(k)} - \boldsymbol{x}_i\|_p^{1-p} |y_{(k)}^{(t)} - x_i^{(t)}|^{p-2}}, \end{aligned} \tag{5}$$

where $y_{(k)}^{(t)}$ denotes the t-th dimension of the k-th iterate. **When $\boldsymbol{y}_{(k)}$ hits the singular set $\mathcal{S}_p$, pNWA cannot solve it but just terminates at $\boldsymbol{y}_{(k)}$.**

2.2 q-th Power Weiszfeld Algorithm without Singularity

Lai et al. (2024) propose a q-th power Weiszfeld algorithm without singularity (qPWAWS) to handle the singularity problem in the special case $p = 2, 1 \leqslant q < 2$:

$$\boldsymbol{y}_{(k+1)} = \begin{cases} \mathbf{T}_{2,q}(\boldsymbol{y}_{(k)}) := \frac{\sum_{i=1}^m \xi_i \|\boldsymbol{y}_{(k)} - \boldsymbol{x}_i\|_2^{q-2} \boldsymbol{x}_i}{\sum_{i=1}^m \xi_i \|\boldsymbol{y}_{(k)} - \boldsymbol{x}_i\|_2^{q-2}} \\ \qquad \text{if } \boldsymbol{y}_{(k)} \notin \{\boldsymbol{x}_i\}_{i=1}^m, \\ \mathbf{T}_s(\boldsymbol{y}_{(k)}) := \boldsymbol{y}_{(k)} - \lambda_* \nabla D_{2,q}(\boldsymbol{y}_{(k)}) \\ \qquad \text{if } \boldsymbol{y}_{(k)} = \boldsymbol{x}_l \text{ for some } l \in \{1, \ldots, m\}, \end{cases} \tag{6}$$

where

$$\nabla D_{2,q}(\boldsymbol{y}_{(k)}) = \sum_{i \neq l} q \xi_i \|\boldsymbol{y}_{(k)} - \mathbf{x}_i\|_2^{q-2} (\boldsymbol{y}_{(k)} - \mathbf{x}_i) \tag{7}$$

is the q-th-powered ℓ_2-norm de-singularity subgradient and $\lambda_* > 0$. Intuitively, $\nabla D_{2,q}$ removes the singular component corresponding to $\boldsymbol{x}_l$ and serves as a conventional gradient in the subgradient descent step $\mathbf{T}_s$.

qPWAWS can escape from each singular point and will not revisit it again. Since there are finite singular points when $p = 2$, qPWAWS can eventually escape from the singular set $\mathcal{S}_2$ (see Figure 1a). **However, it cannot be directly adopted in the $1 \leqslant p < 2$ case, because the singular set $\mathcal{S}_p$ is a continuum and qPWAWS may not eventually escape from $\mathcal{S}_p$. Moreover, the step size λ_* in $\mathbf{T}_s$ cannot be uniformly chosen with respect to (w.r.t.) $\mathcal{S}_p$ due to its infinite elements. The characterizations of the subgradients, minimum and descent direction are also difficult. These three problems affect the convergence property of qPWAWS.** Therefore, a new methodology should be developed to overcome these new challenges for the $1 \leqslant p < 2$ case.

3 q-th-Powered ℓ_p-Norm Weiszfeld Algorithm without Singularity

In this section, we present qPpNWAWS for solving qPpNWLP with $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$. For a convenient illustration, we let $\eta_i := \xi_i^{\frac{1}{q}}$ and reformulate (1) as

$$\boldsymbol{x}_* \in \operatorname*{arg\,min}_{\boldsymbol{y}\in\mathbb{R}^d} C_{p,q}(\boldsymbol{y}) := \sum_{i=1}^{m} \eta_i^q \|\boldsymbol{y} - \boldsymbol{x}_i\|_p^q. \tag{8}$$

$C_{p,q}$ is strictly convex and there exists a unique minimum point $\boldsymbol{x}_*$ for (8) with $1 < p < 2$, while $C_{p,q}$ is convex and there exists one or more minimum points for (8) with $p = 1$. The construction of qPpNWAWS consists of 4 steps:

1. The update at a nonsingular iterate $\boldsymbol{y}_{(k)} \notin \mathcal{S}_p$ is designed as a Weiszfeld-style one, which makes $C_{p,q}(\boldsymbol{y}_{(k)})$ nonincreasing.
2. Define the q-th-powered ℓ_p-norm de-singularity subgradient $\nabla D_{p,q}(\boldsymbol{y})$ for $\boldsymbol{y} \in \mathcal{S}_p$ to characterize the subgradients of $C_{p,q}(\boldsymbol{y})$ and the minimum of $C_{p,q}$.
3. Adopt $\nabla D_{p,q}(\boldsymbol{y}_{(k)})$ to construct the iterative update at the singular iterate $\boldsymbol{y}_{(k)} \in \mathcal{S}_p$ to reduce the cost function.
4. Establish the convergence proof for qPpNWAWS.

3.1 Update Formula at Nonsingular Iterates

First, we consider the simplest case where the current iterate $\boldsymbol{y}_{(k)} \notin \mathcal{S}_p$. The corresponding update formula is:

$$y_{(k+1)}^{(t)} := \left(\mathbf{T}_{p,q}(\boldsymbol{y}_{(k)})\right)^{(t)} := \frac{\sum_{i=1}^{m} \eta_i^q \|\boldsymbol{y}_{(k)} - \boldsymbol{x}_i\|_p^{q-p} |y_{(k)}^{(t)} - x_i^{(t)}|^{p-2} x_i^{(t)}}{\sum_{i=1}^{m} \eta_i^q \|\boldsymbol{y}_{(k)} - \boldsymbol{x}_i\|_p^{q-p} |y_{(k)}^{(t)} - x_i^{(t)}|^{p-2}}. \tag{9}$$

The following descent property guarantees that (9) will reduce $C_{p,q}$ at any non-minimum nonsingular iterate.

Theorem 1 (Descent Property at Nonsingular Iterates)**.** *Let the cost function $C_{p,q}$ and the operator $\mathbf{T}_{p,q}$ be defined in (8) and (9), respectively. For $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$, if $\boldsymbol{y}_{(k)} \notin \mathcal{S}_p$, then $C_{p,q}(\mathbf{T}_{p,q}(\boldsymbol{y}_{(k)})) \leqslant C_{p,q}(\boldsymbol{y}_{(k)})$ with equality holds only when $\mathbf{T}_{p,q}(\boldsymbol{y}_{(k)}) = \boldsymbol{y}_{(k)}$.*

The proof is provided in Supplementary B.1. The following corollary characterizes a minimum point $\boldsymbol{x}_*$ of (8) if $\boldsymbol{x}_* \notin \mathcal{S}_p$.

Corollary 2. *If $\boldsymbol{y}_{(k)} \notin \mathcal{S}_p$, then $\mathbf{T}_{p,q}(\boldsymbol{y}_{(k)}) = \boldsymbol{y}_{(k)} \Leftrightarrow \boldsymbol{y}_{(k)}$ is a minimum point of model (8), i.e., $\boldsymbol{y}_{(k)} = \boldsymbol{x}_*$.*

The proof is provided in Supplementary B.2. We then turn to the singular case.

3.2 Characterization of Subgradients and Minimum

Before we derive the iterative update for the singular iterate $\boldsymbol{y}_{(k)} \in \mathcal{S}_p$, we first introduce the de-singularity subgradient of $C_{p,q}(\boldsymbol{y}_{(k)})$ and characterize the minimum point(s) of (8).

Definition 3 (Subgradient, Rockafellar and Wets 2009)**.** *Let $C_{p,q} : \mathbb{R}^d \to \mathbb{R}$ be a convex function. A vector $\boldsymbol{v} \in \mathbb{R}^d$ is called a subgradient of $C_{p,q}$ at $\boldsymbol{y} \in \mathbb{R}^d$ if for all $\boldsymbol{z} \in \mathbb{R}^d$,*

$$C_{p,q}(\boldsymbol{z}) - C_{p,q}(\boldsymbol{y}) \geqslant \boldsymbol{v}^\top(\boldsymbol{z} - \boldsymbol{y}). \tag{10}$$

The set of all subgradients at $\boldsymbol{y}$ is denoted by $\partial C_{p,q}(\boldsymbol{y})$. If $C_{p,q}$ is differentiable at $\boldsymbol{y}$, then $\partial C_{p,q}(\boldsymbol{y})$ reduces to the gradient $\nabla C_{p,q}(\boldsymbol{y})$.

To construct $\partial C_{p,q}(\boldsymbol{y}_{(k)})$, we need to identify the singular component(s) of $\boldsymbol{y}_{(k)}$.

Definition 4 (Singular Component(s))**.** *For $\boldsymbol{y} \in \mathcal{S}_p$, each $t \in \{1, \ldots, d\}$ and each $i \in \{1, 2, \ldots, m\}$, let*

$$U_i(\boldsymbol{y}) := \{t \in \{1, \ldots, d\} : y^{(t)} = x_i^{(t)}\}, \tag{11}$$

$$V_t(\boldsymbol{y}) := \{i \in \{1, \ldots, m\} : y^{(t)} = x_i^{(t)}\}, \tag{12}$$

which represent the index sets of the dimensions and the data points such that $y^{(t)} = x_i^{(t)}$, respectively.

Figure 2 shows an intuitive example for $U_i(\boldsymbol{y}_{(k)})$ and $V_t(\boldsymbol{y}_{(k)})$.

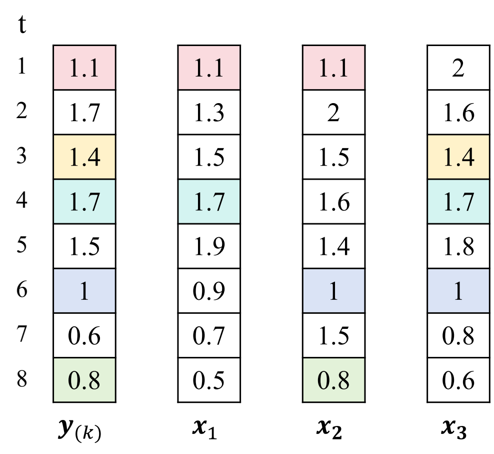


Figure 2: An intuitive example for $U_i(\boldsymbol{y}_{(k)})$ and $V_t(\boldsymbol{y}_{(k)})$: $U_1(\boldsymbol{y}_{(k)}) = \{1, 4\}$, $U_2(\boldsymbol{y}_{(k)}) = \{1, 6, 8\}$, $U_3(\boldsymbol{y}_{(k)}) = \{3, 4, 6\}$, and $V_1(\boldsymbol{y}_{(k)}) = \{1, 2\}$, $V_2(\boldsymbol{y}_{(k)}) = \emptyset$, $V_3(\boldsymbol{y}_{(k)}) = \{3\}$, $V_4(\boldsymbol{y}_{(k)}) = \{1, 3\}$, $V_6(\boldsymbol{y}_{(k)}) = \{2, 3\}$, $V_8(\boldsymbol{y}_{(k)}) = \{2\}$.

Definition 5 (**q-th-Powered ℓ_p-Norm De-singularity Subgradient**)**.** *By removing the singular term(s), we define the de-singularity part $D_{p,q}:\mathbb{R}^d\to\mathbb{R}$ of $C_{p,q}$ and the q-th-powered ℓ_p-norm de-singularity subgradient as follows:*

$$D_{p,q}(\boldsymbol{y}):=\sum_{i=1}^{m}\eta_i^q\Big(\sum_{t\notin U_i(\boldsymbol{y})}|y^{(t)}-x_i^{(t)}|^p\Big)^{\frac{q}{p}},\qquad(13)$$

$$\big(\nabla D_{p,q}(\boldsymbol{y})\big)^{(t)}:=\sum_{i\notin V_t(\boldsymbol{y})}q\eta_i^q\|\boldsymbol{y}-\boldsymbol{x}_i\|_p^{q-p}|y^{(t)}-x_i^{(t)}|^{p-2}(y^{(t)}-x_i^{(t)}).\ (14)$$

Theorem 6 (Characterization of Subgradients and Minimum)**.** *For $\boldsymbol{y}\in\mathcal{S}_p$,*

$$\partial C_{p,q}(\boldsymbol{y})=\begin{cases}\{\nabla D_{1,1}(\boldsymbol{y})+\boldsymbol{u}\}\ \text{where}\ -a^{(t)}\leqslant u^{(t)}\leqslant a^{(t)},\\ \qquad\forall t,\qquad \text{if}\ p=q=1,\\ \{\nabla D_{p,1}(\boldsymbol{y})\},\ \text{if}\ q{=}1,1{<}p{<}2,\\ \qquad\qquad \boldsymbol{y}\in\mathcal{S}_p\backslash\{\boldsymbol{x}_i\}_{i=1}^m,\\ \{\nabla D_{p,1}(\boldsymbol{x}_l)+\eta_l\boldsymbol{b}\}\quad \text{where}\quad \|\boldsymbol{b}\|_r\leqslant 1,\\ \qquad\qquad \text{if}\ q{=}1,1{<}p{<}2,\boldsymbol{y}=\boldsymbol{x}_l,\\ \{\nabla D_{p,q}(\boldsymbol{y})\},\ \text{if}\ 1<q\leqslant p,1<p<2,\end{cases}\qquad(15)$$

where $a^{(t)}=\sum_{i\in V_t(\boldsymbol{y})}\eta_i$ and $\|\cdot\|_r$ is the conjugate norm of $\|\cdot\|_p$ such that $\frac{1}{r}+\frac{1}{p}=1$.

The proof is provided in Supplementary B.3. According to Fermat's rule, $\boldsymbol{y}\in\mathcal{S}_p$ is a minimum point of (8) if and only if $\mathbf{0}_d\in\partial C_{p,q}(\boldsymbol{y})$, which is easy to verify. If $\boldsymbol{y}_{(k)}$ is not a minimum point, the following theorem shows a descent direction of $C_{p,q}(\boldsymbol{y}_{(k)})$.

Theorem 7 (Descent Property at Singular Iterates)**.** *For $1\leqslant p<2$ and $1\leqslant q\leqslant p$, define the following direction*

$$\mathscr{D}_{p,q}(\boldsymbol{y})=\begin{cases}\big(\nabla D_{p,1}(\boldsymbol{x}_l)\big)^{\frac{r}{p}}\ \ \text{where}\ \frac{1}{r}+\frac{1}{p}=1,\\ \qquad\qquad \text{if}\ q{=}1,1{<}p{<}2,\boldsymbol{y}=\boldsymbol{x}_l,\\ \nabla D_{p,q}(\boldsymbol{y}),\qquad\qquad \text{else},\end{cases}\qquad(16)$$

where $(\cdot)^{\frac{r}{p}}$ denotes the element-wise signed power. If $\boldsymbol{y}\in\mathcal{S}_p$ is not a minimum point of (8), then there exists some $\lambda_*>0$ such that for any $0<\lambda\leqslant\lambda_*$, $C_{p,q}(\boldsymbol{y}-\lambda\mathscr{D}_{p,q}(\boldsymbol{y}))<C_{p,q}(\boldsymbol{y})$.

The proof is provided in Supplementary B.4. The key point is to verify that $-\mathscr{D}_{p,q}(\boldsymbol{y})$ is a descent direction if $\boldsymbol{y}$ is not a minimum point. To determine the step size λ_* in practice, we can start with an initial value of $\lambda_0=\|\mathscr{D}_{p,q}(\boldsymbol{y}_{(k)})\|_p$ and implement a line search $\lambda_{w+1}=\rho\lambda_w$ with $0<\rho<1$, until we find a value of λ_* such that $C_{p,q}(\boldsymbol{y}_{(k)}-\lambda_*\mathscr{D}_{p,q}(\boldsymbol{y}_{(k)}))<C_{p,q}(\boldsymbol{y}_{(k)})$. Then we can construct the iterative update at the singular point $\boldsymbol{y}_{(k)}$ as

$$\boldsymbol{y}_{(k+1)}:=\mathbf{T}_s(\boldsymbol{y}_{(k)}):=\boldsymbol{y}_{(k)}-\lambda_*\mathscr{D}_{p,q}(\boldsymbol{y}_{(k)}).\qquad(17)$$

Combining (9) and (17), the whole qPpNWAWS is given by:

$$\boldsymbol{y}_{(k+1)}:=\mathbf{T}(\boldsymbol{y}_{(k)}):=\begin{cases}\mathbf{T}_{p,q}(\boldsymbol{y}_{(k)}) & \text{if}\ \boldsymbol{y}_{(k)}\notin\mathcal{S}_p,\\ \mathbf{T}_s(\boldsymbol{y}_{(k)}) & \text{if}\ \boldsymbol{y}_{(k)}\in\mathcal{S}_p.\end{cases}\qquad(18)$$

Theorems 1 and 7 indicate that $C_{p,q}(\boldsymbol{y}_{(k+1)})<C_{p,q}(\boldsymbol{y}_{(k)})$ if $\boldsymbol{y}_{(k)}$ is not a minimum point (which can be characterized by qPpNWAWS). Moreover, since $C_{p,q}(\boldsymbol{y})\geqslant 0$ for all $\boldsymbol{y}\in\mathbb{R}^d$, we can conclude that the sequence $\{C_{p,q}(\boldsymbol{y}_{(k)})\}_{k\in\mathbb{N}}$ converges.

3.3 Convergence Theorem

To analyze the convergence of qPpNWAWS, we first provide several lemmas on some properties of the operators involved. The first lemma indicates that the operator $\mathbf{T}_{p,q}$ designed for the nonsingular iterates can be extended to the singular iterates.

Lemma 8. *Let the operator $\mathbf{T}_{p,q}$ be defined in (9). Then for $1\leqslant q\leqslant p$, $1\leqslant p<2$,*

$$\lim_{\boldsymbol{y}\to\boldsymbol{x},\boldsymbol{y}\notin\mathcal{S}_p}\mathbf{T}_{p,q}(\boldsymbol{y})=\boldsymbol{x},\quad\forall\boldsymbol{x}\in\mathcal{S}_p.\qquad(19)$$

The proof is provided in Supplementary B.5. Therefore, we can define

$$\mathbf{T}_{p,q}(\boldsymbol{x}):=\boldsymbol{x},\quad\forall\boldsymbol{x}\in\mathcal{S}_p,\ 1\leqslant q\leqslant p,\ 1\leqslant p<2.\quad(20)$$

The second lemma confirms that the sequence $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ generated by qPpNWAWS is bounded, which is based on the descent property of $\{C_{p,q}(\boldsymbol{y}_{(k)})\}_{k\in\mathbb{N}}$.

Lemma 9. *The sequence $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ generated by qPpNWAWS is bounded.*

The proof is provided in Supplementary B.6. As emphasized above, the singular set $\mathcal{S}_p$ contains infinite points in the case $1\leqslant p<2$, hence the operator $\mathbf{T}_{p,q}$ may not escape from $\mathcal{S}_p$. This is a major unavoidable obstacle for convergence analysis. To overcome this obstacle, we combine Lemma 9 and the Bolzano-Weierstrasz theorem, then there exists at least one limit point y_* and a subsequence $\{\boldsymbol{y}_{(k_v)}\}_{v\in\mathbb{N}}$ such that $\lim_{v\to\infty}\boldsymbol{y}_{(k_v)}=\boldsymbol{y}_*$. The following lemma indicates that there are only a finite number of limit points of $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ in the strictly convex case $1<p<2$, even though $\mathcal{S}_p$ has infinite points. This is a novel and significant theoretical result that plays a crucial part in the convergence proof.

Lemma 10. *If $1<p<2$, there are only a finite number of limit points of $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$. Moreover, all these limit points have the same cost function value.*

The proof is provided in Supplementary B.7.

Theorem 11 (Convergence Theorem)**.** *Let $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ be the iteration sequence generated by qPpNWAWS in (18). If $\boldsymbol{y}_{(k)}$ hits the minimum point $\boldsymbol{x}_*$ of model (8), the characterization of minimum (Corollary 2 and Theorem 6) ensures that this could be recognized and the algorithm will be stopped. Otherwise, the cost function sequence $\{C_{p,q}(\boldsymbol{y}_{(k)})\}_{k\in\mathbb{N}}$ converges. Assume $1<p<2$ in addition. If $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ hits $\mathcal{S}_p$ for a finite number of times, then $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ also converges. On the other hand, if $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ has a nonsingular cluster point, then this cluster point is exactly the minimum point $\boldsymbol{x}_*$ and the entire sequence $\{\boldsymbol{y}_{(k)}\}_{k\in\mathbb{N}}$ converges to $\boldsymbol{x}_*$.*

The proof is provided in Supplementary B.8. We provide a practical way to verify the conditions of this theorem. When the algorithm reaches the convergence tolerance, we can check whether the last few iterates are singular points. If not, we can consider that $y_{(k)}$ converges to exactly the minimum point $\boldsymbol{x}_*$.

Data Set	Region	Time	Periods	Frequency	# Assets
CSI300	CN	Mar/16/2015- May/19/2017	534	Daily	47
NYSE(N)	US	Jan/1/1985 - Jun/30/2010	6431	Daily	23
FTSE100	UK	Nov/07/2002 - Nov/04/2016	717	Weekly	83
NASDAQ100	US	Mar/11/2004 - Nov/04/2016	596	Weekly	82
FF100	US	Jul/1971 - May/2023	623	Monthly	100
FF100MEOP	US	Jul/1971 - May/2023	623	Monthly	100

Table 1: Profiles of six benchmark data sets.

q \ p	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	4.92 ± 0.29	2.56 ± 1.07	2.15 ± 0.42	2.08 ± 0.36	2.04 ± 0.30	2.02 ± 0.28	1.99 ± 0.27	1.98 ± 0.25	1.97 ± 0.24	1.97 ± 0.24
1.1	-	2.83 ± 1.27	2.11 ± 0.39	2.07 ± 0.33	2.02 ± 0.29	2.00 ± 0.26	1.99 ± 0.26	1.97 ± 0.24	1.97 ± 0.24	1.96 ± 0.23
1.2	-	-	2.24 ± 0.63	2.04 ± 0.30	2.01 ± 0.27	2.00 ± 0.25	1.98 ± 0.24	1.97 ± 0.23	1.96 ± 0.23	1.96 ± 0.23
1.3	-	-	-	2.05 ± 0.36	2.00 ± 0.26	1.98 ± 0.24	1.97 ± 0.22	1.96 ± 0.22	1.96 ± 0.23	1.96 ± 0.23
1.4	-	-	-	-	2.00 ± 0.25	1.98 ± 0.23	1.96 ± 0.22	1.96 ± 0.22	1.96 ± 0.23	1.95 ± 0.23
1.5	-	-	-	-	-	1.98 ± 0.22	1.96 ± 0.23	1.96 ± 0.23	1.96 ± 0.22	1.95 ± 0.22
1.6	-	-	-	-	-	-	1.96 ± 0.23	1.96 ± 0.22	1.95 ± 0.22	1.95 ± 0.22
1.7	-	-	-	-	-	-	-	1.96 ± 0.21	1.95 ± 0.22	1.95 ± 0.22
1.8	-	-	-	-	-	-	-	-	1.95 ± 0.22	1.95 ± 0.22
1.9	-	-	-	-	-	-	-	-	-	1.95 ± 0.22

Table 2: Average number of iterates for qPpNWAWS to reduce the cost function at a singular point on CSI300 (mean±STD).

q \ p	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	3.91 ± 0.60	1.90 ± 0.98	1.74 ± 0.81	1.68 ± 0.78	1.64 ± 0.74	1.60 ± 0.68	1.57 ± 0.64	1.54 ± 0.61	1.53 ± 0.60	1.52 ± 0.59
1.1	-	2.07 ± 1.16	1.72 ± 0.81	1.67 ± 0.78	1.63 ± 0.73	1.60 ± 0.70	1.57 ± 0.66	1.54 ± 0.61	1.53 ± 0.60	1.52 ± 0.59
1.2	-	-	1.75 ± 0.86	1.66 ± 0.77	1.62 ± 0.72	1.59 ± 0.68	1.56 ± 0.64	1.53 ± 0.61	1.52 ± 0.59	1.51 ± 0.58
1.3	-	-	-	1.65 ± 0.77	1.61 ± 0.70	1.58 ± 0.66	1.55 ± 0.63	1.53 ± 0.60	1.51 ± 0.58	1.50 ± 0.57
1.4	-	-	-	-	1.60 ± 0.68	1.57 ± 0.66	1.55 ± 0.63	1.52 ± 0.59	1.50 ± 0.57	1.49 ± 0.55
1.5	-	-	-	-	-	1.57 ± 0.65	1.55 ± 0.63	1.51 ± 0.58	1.49 ± 0.55	1.47 ± 0.53
1.6	-	-	-	-	-	-	1.54 ± 0.63	1.50 ± 0.57	1.47 ± 0.53	1.46 ± 0.51
1.7	-	-	-	-	-	-	-	1.48 ± 0.54	1.46 ± 0.50	1.45 ± 0.50
1.8	-	-	-	-	-	-	-	-	1.46 ± 0.50	1.45 ± 0.50
1.9	-	-	-	-	-	-	-	-	-	1.45 ± 0.50

Table 3: Average number of iterates for qPpNWAWS to reduce the cost function at a singular point on NYSE(N) (mean±STD).

q \ p		1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	*Time*	0.0271	0.0193	0.0180	0.0169	0.0161	0.0155	0.0148	0.0143	0.0135	0.0130
	Iter	35.93 ± 15.04	15.02 ± 2.79	13.89 ± 2.07	13.38 ± 2.49	12.83 ± 2.68	12.34 ± 2.78	11.90 ± 2.82	11.51 ± 2.83	11.19 ± 2.81	10.92 ± 2.83
1.1	*Time*	-	0.0184	0.0173	0.0162	0.0153	0.0144	0.0136	0.0131	0.0123	0.0118
	Iter	-	15.01 ± 2.37	13.63 ± 1.89	12.95 ± 2.14	12.28 ± 2.25	11.73 ± 2.35	11.27 ± 2.37	10.84 ± 2.37	10.51 ± 2.37	10.24 ± 2.36
1.2	*Time*	-	-	0.0170	0.0157	0.0145	0.0136	0.0127	0.0122	0.0114	0.0109
	Iter	-	-	13.65 ± 1.86	12.66 ± 1.89	11.86 ± 1.87	11.27 ± 1.96	10.75 ± 1.98	10.32 ± 2.01	10.01 ± 2.03	9.74 ± 2.03
1.3	*Time*	-	-	-	0.0151	0.0140	0.0128	0.0119	0.0114	0.0107	0.0102
	Iter	-	-	-	12.45 ± 1.67	11.52 ± 1.55	10.82 ± 1.65	10.30 ± 1.64	9.90 ± 1.69	9.60 ± 1.74	9.32 ± 1.73
1.4	*Time*	-	-	-	-	0.0135	0.0122	0.0113	0.0107	0.0100	0.0095
	Iter	-	-	-	-	11.20 ± 1.24	10.44 ± 1.34	9.91 ± 1.38	9.48 ± 1.40	9.18 ± 1.48	8.90 ± 1.48
1.5	*Time*	-	-	-	-	-	0.0114	0.0109	0.0100	0.0092	0.0088
	Iter	-	-	-	-	-	10.09 ± 1.04	9.55 ± 1.09	9.12 ± 1.15	8.75 ± 1.24	8.49 ± 1.27
1.6	*Time*	-	-	-	-	-	-	0.0101	0.0094	0.0086	0.0081
	Iter	-	-	-	-	-	-	9.25 ± 0.86	8.77 ± 0.91	8.38 ± 1.05	8.08 ± 1.09
1.7	*Time*	-	-	-	-	-	-	-	0.0086	0.0079	0.0073
	Iter	-	-	-	-	-	-	-	8.46 ± 0.73	7.97 ± 0.88	7.63 ± 1.02
1.8	*Time*	-	-	-	-	-	-	-	-	0.0072	0.0070
	Iter	-	-	-	-	-	-	-	-	7.66 ± 0.71	7.45 ± 0.69
1.9	*Time*	-	-	-	-	-	-	-	-	-	0.0064
	Iter	-	-	-	-	-	-	-	-	-	7.19 ± 0.41

Table 4: Average computational time (in seconds) and average number of iterations (mean±STD) for qPpNWAWS on CSI300.

To end this section, we indicate that a general constant-step subgradient descent method cannot guarantee convergence in the objective value for qPpNWLP due to the varying λ_* in Theorem 7. Besides, $\nabla C_{p,q}(\boldsymbol{y})$ in (2) may not be Lipschitz continuous even for $\boldsymbol{y} \notin \mathcal{S}_p$. We raise a counterexample: let $C_{p,q}(\boldsymbol{y}) := \|\boldsymbol{y}\|_p^q$, $\boldsymbol{y}_1 = \varepsilon \mathbf{1}_d > \mathbf{0}_d$, and $\boldsymbol{y}_2 = \frac{\varepsilon}{2}\mathbf{1}_d$. Then

$$\frac{\|\nabla C_{p,q}(\boldsymbol{y}_1) - \nabla C_{p,q}(\boldsymbol{y}_2)\|_2}{\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2} = (2 - 2^{2-q}) q d^{\frac{q-p}{p}} \varepsilon^{q-2}. \tag{21}$$

q \ p		1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	*Time*	0.0011	0.0050	0.0071	0.0074	0.0060	0.0064	0.0055	0.0035	0.0033	0.0032
	Iter	26.71 ± 15.93	14.84 ± 6.67	13.62 ± 4.51	12.72 ± 3.35	12.07 ± 2.86	11.50 ± 2.73	11.02 ± 2.54	10.62 ± 2.45	10.29 ± 2.44	10.03 ± 2.43
1.1	*Time*	-	0.0067	0.0066	0.0068	0.0052	0.0061	0.0051	0.0032	0.0031	0.0029
	Iter	-	14.70 ± 8.64	13.29 ± 4.22	12.29 ± 3.04	11.56 ± 2.55	10.99 ± 2.34	10.52 ± 2.22	10.09 ± 2.11	9.77 ± 2.11	9.52 ± 2.10
1.2	*Time*	-	-	0.0075	0.0059	0.0044	0.0058	0.0048	0.0030	0.0029	0.0027
	Iter	-	-	13.05 ± 3.99	11.96 ± 2.84	11.13 ± 2.27	10.53 ± 2.06	10.06 ± 1.95	9.64 ± 1.85	9.33 ± 1.87	9.08 ± 1.90
1.3	*Time*	-	-	-	0.0056	0.0051	0.0057	0.0041	0.0029	0.0027	0.0026
	Iter	-	-	-	11.70 ± 2.75	10.77 ± 1.98	10.13 ± 1.80	9.65 ± 1.70	9.24 ± 1.64	8.92 ± 1.68	8.67 ± 1.72
1.4	*Time*	-	-	-	-	0.0048	0.0052	0.0028	0.0027	0.0025	0.0024
	Iter	-	-	-	-	10.47 ± 1.85	9.78 ± 1.58	9.29 ± 1.54	8.86 ± 1.47	8.55 ± 1.53	8.29 ± 1.57
1.5	*Time*	-	-	-	-	-	0.0043	0.0026	0.0025	0.0024	0.0023
	Iter	-	-	-	-	-	9.47 ± 1.40	8.96 ± 1.43	8.52 ± 1.36	8.17 ± 1.40	7.89 ± 1.46
1.6	*Time*	-	-	-	-	-	-	0.0025	0.0024	0.0022	0.0021
	Iter	-	-	-	-	-	-	8.66 ± 1.37	8.17 ± 1.27	7.80 ± 1.34	7.52 ± 1.33
1.7	*Time*	-	-	-	-	-	-	-	0.0022	0.0020	0.0019
	Iter	-	-	-	-	-	-	-	7.83 ± 1.16	7.38 ± 1.02	7.10 ± 1.10
1.8	*Time*	-	-	-	-	-	-	-	-	0.0019	0.0018
	Iter	-	-	-	-	-	-	-	-	7.07 ± 0.97	6.84 ± 1.03
1.9	*Time*	-	-	-	-	-	-	-	-	-	0.0016
	Iter	-	-	-	-	-	-	-	-	-	6.50 ± 0.89

Table 5: Average computational time (in seconds) and average number of iterations (mean±STD) for qPpNWAWS on NYSE(N).

q \ p	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	0.80 ± 0.07	0.64 ± 0.06	0.61 ± 0.06	0.58 ± 0.07	0.55 ± 0.08	0.52 ± 0.09	0.49 ± 0.10	0.47 ± 0.10	0.46 ± 0.11	0.44 ± 0.12
1.1	-	0.63 ± 0.05	0.60 ± 0.05	0.56 ± 0.06	0.52 ± 0.06	0.49 ± 0.07	0.46 ± 0.08	0.44 ± 0.09	0.42 ± 0.10	0.40 ± 0.11
1.2	-	-	0.59 ± 0.05	0.55 ± 0.05	0.51 ± 0.05	0.47 ± 0.06	0.44 ± 0.07	0.41 ± 0.08	0.39 ± 0.09	0.37 ± 0.10
1.3	-	-	-	0.54 ± 0.05	0.50 ± 0.04	0.45 ± 0.05	0.41 ± 0.06	0.38 ± 0.07	0.36 ± 0.08	0.34 ± 0.09
1.4	-	-	-	-	0.48 ± 0.04	0.43 ± 0.04	0.39 ± 0.05	0.35 ± 0.06	0.32 ± 0.07	0.30 ± 0.08
1.5	-	-	-	-	-	0.41 ± 0.03	0.36 ± 0.04	0.32 ± 0.05	0.29 ± 0.06	0.26 ± 0.07
1.6	-	-	-	-	-	-	0.34 ± 0.03	0.29 ± 0.03	0.25 ± 0.05	0.22 ± 0.08
1.7	-	-	-	-	-	-	-	0.26 ± 0.03	0.21 ± 0.04	0.17 ± 0.04
1.8	-	-	-	-	-	-	-	-	0.18 ± 0.02	0.14 ± 0.02
1.9	-	-	-	-	-	-	-	-	-	0.09 ± 0.01

Table 6: Average computational convergence rate (mean±STD) for qPpNWAWS on CSI300.

q \ p	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	0.75 ± 0.10	0.64 ± 0.07	0.61 ± 0.08	0.58 ± 0.08	0.55 ± 0.08	0.53 ± 0.09	0.50 ± 0.10	0.48 ± 0.10	0.46 ± 0.10	0.45 ± 0.11
1.1	-	0.63 ± 0.07	0.60 ± 0.07	0.57 ± 0.08	0.53 ± 0.08	0.50 ± 0.08	0.47 ± 0.09	0.45 ± 0.09	0.43 ± 0.10	0.42 ± 0.10
1.2	-	-	0.59 ± 0.07	0.55 ± 0.07	0.51 ± 0.07	0.48 ± 0.08	0.45 ± 0.08	0.42 ± 0.08	0.40 ± 0.09	0.38 ± 0.10
1.3	-	-	-	0.54 ± 0.07	0.50 ± 0.07	0.46 ± 0.07	0.42 ± 0.07	0.39 ± 0.08	0.37 ± 0.09	0.35 ± 0.10
1.4	-	-	-	-	0.48 ± 0.07	0.44 ± 0.07	0.40 ± 0.07	0.36 ± 0.07	0.34 ± 0.09	0.32 ± 0.11
1.5	-	-	-	-	-	0.42 ± 0.06	0.37 ± 0.07	0.33 ± 0.07	0.31 ± 0.09	0.28 ± 0.11
1.6	-	-	-	-	-	-	0.35 ± 0.08	0.30 ± 0.08	0.27 ± 0.10	0.24 ± 0.13
1.7	-	-	-	-	-	-	-	0.27 ± 0.09	0.23 ± 0.11	0.19 ± 0.07
1.8	-	-	-	-	-	-	-	-	0.18 ± 0.07	0.15 ± 0.07
1.9	-	-	-	-	-	-	-	-	-	0.11 ± 0.08

Table 7: Average computational convergence rate (mean±STD) for qPpNWAWS on NYSE(N).

Since $q<2$, (21) implies $\lim_{\varepsilon\to 0} \frac{\|\nabla C_{p,q}(\boldsymbol{y}_1)-\nabla C_{p,q}(\boldsymbol{y}_2)\|_2}{\|\boldsymbol{y}_1-\boldsymbol{y}_2\|_2} = \infty$. Therefore, a general subgradient descent method cannot achieve the sublinear convergence rate $O(\frac{1}{k})$. However, practical experiments show that qPpNWAWS achieves a linear computational convergence rate, which is efficient to solve qPpNWLP.

The whole procedure of qPpNWAWS is illustrated in Supplementary A. In each iteration, the algorithm first identifies whether the current iterate is a singular point and the dimensions where the singularity occurs. Next, it computes the de-singularity subgradient (or the normal gradient if there is no singularity). The de-singularity subgradient can then be used to determine whether the current iterate is a minimum point. If it is not, the algorithm performs a de-singularity subgradient descent step to reduce the cost function, proceeding to the next iteration until the convergence tolerance is met.

4 Experiment Result

We adopt the evaluating baseline in (Lai et al. 2024) with tests from different aspects to assess the performance of the proposed qPpNWAWS. In the median reversion strategy (Huang et al. 2016) for online portfolio selection (**OPS**, Li, Sahoo, and Hoi 2016; Lai et al. 2018c, 2020; Lai and Yang 2023), an important step is to compute the median of the asset prices in a recent time window. In the context of this

q \ p		1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	CW	2.0603	1.8979	1.9468	1.9223	1.9012	2.0409	**2.0932**	2.0550	2.0468	1.9474
	SR	0.0563	0.0523	0.0538	0.0532	0.0523	0.0561	**0.0574**	0.0563	0.0560	0.0531
1.1	CW	-	1.9107	2.0278	1.8901	1.9175	2.0327	2.0157	2.0334	2.0160	1.9007
	SR	-	0.0526	0.0561	0.0520	0.0527	0.0559	0.0553	0.0557	0.0552	0.0518
1.2	CW	-	-	1.9781	2.0071	1.9788	2.0842	2.0296	2.0181	1.9821	1.8649
	SR	-	-	0.0545	0.0554	0.0545	0.0574	0.0557	0.0553	0.0542	0.0508
1.3	CW	-	-	-	1.9891	1.9682	2.0375	2.0247	1.9860	1.8868	1.7689
	SR	-	-	-	0.0548	0.0542	0.0560	0.0556	0.0544	0.0514	0.0477
1.4	CW	-	-	-	-	1.8620	1.9303	1.9373	1.9106	1.8795	1.7889
	SR	-	-	-	-	0.0509	0.0529	0.0530	0.0521	0.0512	0.0484
1.5	CW	-	-	-	-	-	1.7826	1.7957	1.8503	1.8596	1.8244
	SR	-	-	-	-	-	0.0483	0.0487	0.0503	0.0506	0.0495
1.6	CW	-	-	-	-	-	-	1.7073	1.7543	1.7665	1.8125
	SR	-	-	-	-	-	-	0.0457	0.0473	0.0477	0.0492
1.7	CW	-	-	-	-	-	-	-	1.6923	1.7183	1.7589
	SR	-	-	-	-	-	-	-	0.0452	0.0461	0.0475
1.8	CW	-	-	-	-	-	-	-	-	1.6924	1.7223
	SR	-	-	-	-	-	-	-	-	0.0453	0.0463
1.9	CW	-	-	-	-	-	-	-	-	-	1.7101
	SR	-	-	-	-	-	-	-	-	-	0.0459

Table 8: Cumulative wealth (CW) and Sharpe Ratio (SR) of qPpNWAWS on CSI300. The CW and SR for the original setting $(q, p) = (1, 2)$ are 1.7750 and 0.0479, respectively.

q \ p		1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
1.0	CW	$2.0561E+07$	$4.3557E+07$	$6.3893E+07$	$1.7701E+08$	$1.4948E+08$	$2.6522E+08$	$1.8973E+08$	$3.0391E+08$	$3.2967E+08$	$4.1537E+08$
	SR	0.0935	0.0968	0.0980	0.1022	0.1012	0.1036	0.1020	0.1042	0.1042	0.1046
1.1	CW	-	$6.5105E+07$	$1.0056E+08$	$1.9101E+08$	$1.3263E+08$	$2.5663E+08$	$2.5658E+08$	$3.0408E+08$	$4.5008E+08$	$7.1092E+08$
	SR	-	0.0987	0.1002	0.1027	0.1007	0.1034	0.1033	0.1039	0.1051	0.1068
1.2	CW	-	-	$1.4552E+08$	$1.1925E+08$	$1.6110E+08$	$1.5340E+08$	$1.9023E+08$	$3.1231E+08$	$6.0239E+08$	$8.0655E+08$
	SR	-	-	0.1020	0.1006	0.1015	0.1009	0.1017	0.1039	0.1060	0.1072
1.3	CW	-	-	-	$1.1036E+08$	$1.0286E+08$	$1.4315E+08$	$1.7280E+08$	$2.4992E+08$	$6.5705E+08$	$\mathbf{8.5677E+08}$
	SR	-	-	-	0.0998	0.0994	0.1005	0.1012	0.1026	0.1064	**0.1075**
1.4	CW	-	-	-	-	$1.1330E+08$	$1.3560E+08$	$1.6290E+08$	$1.7457E+08$	$5.0738E+08$	$7.7795E+08$
	SR	-	-	-	-	0.0998	0.1001	0.1007	0.1009	0.1053	0.1071
1.5	CW	-	-	-	-	-	$2.2089E+08$	$1.2184E+08$	$2.1543E+08$	$4.6540E+08$	$6.5877E+08$
	SR	-	-	-	-	-	0.1024	0.0993	0.1018	0.1049	0.1064
1.6	CW	-	-	-	-	-	-	$7.5558E+07$	$2.3410E+08$	$3.4107E+08$	$5.0696E+08$
	SR	-	-	-	-	-	-	0.0973	0.1021	0.1035	0.1052
1.7	CW	-	-	-	-	-	-	-	$1.2833E+08$	$3.0709E+08$	$5.0657E+08$
	SR	-	-	-	-	-	-	-	0.0993	0.1030	0.1051
1.8	CW	-	-	-	-	-	-	-	-	$2.8561E+08$	$4.0895E+08$
	SR	-	-	-	-	-	-	-	-	0.1026	0.1041
1.9	CW	-	-	-	-	-	-	-	-	-	$3.2349E+08$
	SR	-	-	-	-	-	-	-	-	-	0.1031

Table 9: Cumulative wealth (CW) and Sharpe Ratio (SR) of qPpNWAWS on NYSE(N). The CW and SR for the original setting $(q, p) = (1, 2)$ are $3.3183e + 08$ and 0.1034, respectively.

paper, it aims to find a q-th-powered ℓ_p-norm median

$$\hat{\boldsymbol{x}} \in \arg\min_{\boldsymbol{y} \in \mathbb{R}^d} \sum_{i=1}^{m} \|\boldsymbol{y} - \boldsymbol{x}_i\|_p^q,\ 1 \leqslant q \leqslant p,\ 1 \leqslant p < 2, \quad (22)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ represents the price vector of d assets on the i-th trading day, and $\{\mathbf{x}_i\}_{i=1}^m$ contains the asset prices for the most recent m trading days.

Experiments are conducted on six data sets: CSI300 (Lai et al. 2024), NYSE(N) (Li et al. 2013), FTSE100, NASDAQ100 (Bruni et al. 2016), FF100, and FF100MEOP (Lin et al. 2024). Profiles of these data sets are shown in Table 1. These data sets cover financial markets from different regions like China, the United States, and the United Kingdom. They also cover different frequencies, including daily, weekly, and monthly. Their dimensionalities range from 23 to 100. CSI300 is extracted by Lai et al. (2024) from the CSI300 constituents[1] of Shanghai Stock Exchange and Shenzhen Stock Exchange in China, while FF100 and FF100MEOP are extracted by Lin et al. (2024) from the Kenneth R. French's Data Library[2]. FF100 is built on ME and BE/ME, while FF100MEOP is built on ME and operating profitability, respectively. All these data sets cover a wide range of practical scenarios that are sufficient to test the performance of the proposed qPpNWAWS. Due to the page limit, the experimental results on CSI300 and NYSE(N) are presented in the main text, while those on other data sets are presented in Supplementary C (Tables A1$\sim$A16). All these results consistently show the effectiveness of qPpNWAWS.

The experiments include four parts:

1. We verify that qPpNWAWS successfully reduces the cost function at singular iterates, which solves the singularity problem.
2. We validate the computational efficiency of qPpNWAWS

[1] http://www.csindex.com.cn

[2] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

by analyzing the number of iterations and the running time required for convergence.

3. We verify that the computational convergence rate of qPpNWAWS is a linear convergence rate.
4. By assessing the investing metrics in OPS, we demonstrate the advantages of qPpNWAWS with $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$. Hence qPpNWAWS is useful in a practical sense.

We change p and q in $[1, 1.9]$ with $q \leqslant p$, which covers enough situations of $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$. The time window size m is set as 5 by following previous methods (Huang et al. 2016; Lai et al. 2018a,b, 2022). The convergence tolerance thresholds are set as $Tol = 10^{-4}$ and $Tol\_2 = 10^{-14}$, and the reducing factor ρ in the line search is set as 0.1. As the observation window moves from $t = 1$ to $t = T - m + 1$, there are a total of $(T - m + 1)$ sets of data points $\{\mathbf{x}_i\}_{i=1}^m$. Therefore, we evaluate the average performance of qPpNWAWS by conducting the experiments for $(T - m + 1)$ times on each data set. The experiments are carried out on a desktop workstation with an Intel Core i9-14900KF CPU, 64-GB DDR5 6000-MHz memory cards, and an Nvidia RTX 4080 graphics card with 16-GB independent memory.

4.1 Solving the Singularity Problem

We record the average number of iterations required for qPpNWAWS to successfully reduce the cost function at singular iterates. The starting iterate $\boldsymbol{y}_{(0)}$ is set as the singular point $\boldsymbol{x}_1$. For each (q, p) pair, we calculate the mean and the standard deviation (STD) of the number of iterations required on the $(T - m + 1)$ sets of data points $\{\boldsymbol{x}_i\}_{i=1}^m$, shown in Tables 2 and 3. Results show that qPpNWAWS successfully reduces the cost function in only a few iterations, thereby solving the singularity problem. As p and q increase, the average number of iterations shows a decreasing trend, ranging from 4.92 to 1.95 on CSI300 and from 3.91 to 1.45 on NYSE(N). A smaller ρ may lead to even fewer iterations required in the line search of the step size λ_*.

4.2 Computational Cost and Convergence

We record the average number of iterations and the average running time for qPpNWAWS to achieve convergence in Tables 4 and 5. Results show that qPpNWAWS achieves rapid convergence that the average running times are all smaller than 0.03s and the numbers of iterations are no larger than 36. As p and q increase, the average number of iterations also shows a decreasing trend, ranging from 35.93 to 7.19 on CSI300 and from 26.71 to 6.50 on NYSE(N). To summarize, qPpNWAWS successfully converges at a desirable speed.

4.3 Computational Convergence Rate

We use the following formula to assess the computational convergence rate of qPpNWAWS:

$$\frac{1}{Iter - 2} \sum_{o=3}^{Iter} \frac{\|\boldsymbol{y}_{(o-1)} - \boldsymbol{y}_{(Iter)}\|_2}{\|\boldsymbol{y}_{(o-2)} - \boldsymbol{y}_{(Iter)}\|_2}, \tag{23}$$

where $Iter$ and $\boldsymbol{y}_{(Iter)}$ denote the total number of iterations and the final iterate, respectively. Tables 6 and 7 show the mean and STD of the computational convergence rates for qPpNWAWS with different (q, p) pairs. As p and q increase, the average computational convergence rate decreases from 0.8 to 0.09. Since they are all significantly smaller than 1, qPpNWAWS achieves at least a linear computational convergence rate.

4.4 Investing Performance

To further assess the effectiveness of qPpNWAWS in real-world applications, we employ two main investing metrics, the final cumulative wealth (CW) and the daily Sharpe Ratio (SR, Sharpe 1966), to conduct OPS experiments. The final CW indicates the final gain of an investing strategy at the end of the entire investment, while the SR is a kind of risk-adjusted return. We use qPpNWAWS to compute the q-th-powered ℓ_p-norm median in (22), and then adopt the strategy in (Huang et al. 2016) to produce the CW and SR scores. Results with different (q, p) pairs as well as the original setting $(q, p) = (1, 2)$ in (Huang et al. 2016) are given in Tables 8 and 9. They indicate that qPpNWAWS achieves the best results with $(q, p) = (1, 1.6)$ on CSI300 and with $(q, p) = (1.3, 1.9)$ on NYSE(N). Besides, several (q, p) pairs perform better than the original setting $(q, p) = (1, 2)$. These results indicate that qPpNWAWS for solving qPpNWLP is useful and advantageous with $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$.

5 Conclusions and Future Works

This paper proposes a q-th-Powered ℓ_p-Norm Weiszfeld Algorithm without Singularity (qPpNWAWS) for the q-th-Powered ℓ_p-Norm Weber Location Problem (qPpNWLP) with $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$, which includes all the rest unsolved situations in this problem. One main difficulty to solve this problem is that the singular points constitute a continuum set, so that any gradient-type algorithm may visit the singular set for infinite times. qPpNWAWS is able to characterize the subgradients and minimum at any singular or nonsingular point. If it is not a minimum point, qPpNWAWS can further reduce the cost function. Moreover, it guarantees convergence in the objective function value.

Experimental results on six real-world data sets show that qPpNWAWS successfully reduces the cost function in a few iterations at a singular point. It achieves convergence in a few iterations and shows a linear computational convergence rate. Moreover, it performs well in the online portfolio selection task that its final cumulative wealth and its Sharpe ratio with several (q, p) pairs are higher than those with the original setting $(q, p) = (1, 2)$. Thus qPpNWAWS and qPpNWLP with $1 \leqslant p < 2$ and $1 \leqslant q \leqslant p$ are useful and advantageous in practice. In future works, we will extend the de-singularity methodology to the multi-facility location problem.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grants 62176103, 62276114, 62206110, in part by Guangdong Basic and Applied Basic Research Foundation under grant 2023B1515120064, in part by the Science and Technology Planning Project of Guangzhou under grants 2024A04J9896, 202206030007, in part by Guangdong-Macao Advanced Intelligent Computing Joint Laboratory under grant 2020B1212030003, in part by the Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University under grant 2022LSYS003, in part by the Fundamental Research Funds for the Central Universities, JNU under grant 21623202, and in part by the Major Key Project of PCL (No. PCL2024A04).

References

Aftab, K.; Hartley, R.; and Trumpf, J. 2015. Generalized Weiszfeld Algorithms for L_q Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4): 728–745.

Beckenbach, E. F.; and Bellman, R. 2012. *Inequalities*, volume 30. Springer Science & Business Media.

Brimberg, J.; and Love, R. F. 1993. Global convergence of a generalized iterative procedure for the minisum location problem with lp distances. *Operations Research*, 41(6): 1153–1163.

Bruni, R.; Cesarone, F.; Scozzari, A.; and Tardella, F. 2016. Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models. *Data in Brief*, 8: 858–862.

Chandrasekaran, R.; and Tamir, A. 1989. Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem. *Mathematical Programming*, 44: 293–295.

Chen, R. 1984. Solution of location problems with radial cost functions. *Computers & Mathematics with Applications*, 10(1): 87–94.

Huang, D.; Zhou, J.; Li, B.; Hoi, S. C. H.; and Zhou, S. 2016. Robust Median Reversion Strategy for Online Portfolio Selection. *IEEE Transactions on Knowledge and Data Engineering*, 28(9): 2480–2493.

Lai, Z.-R.; Dai, D.-Q.; Ren, C.-X.; and Huang, K.-K. 2018a. A Peak Price Tracking based Learning System for Portfolio Selection. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7): 2823–2832.

Lai, Z.-R.; Dai, D.-Q.; Ren, C.-X.; and Huang, K.-K. 2018b. Radial basis functions with adaptive input and composite trend representation for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12): 6214–6226.

Lai, Z.-R.; Li, C.; Wu, X.; Guan, Q.; and Fang, L. 2022. Multitrend Conditional Value at Risk for Portfolio Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

Lai, Z.-R.; Tan, L.; Wu, X.; and Fang, L. 2020. Loss Control with Rank-one Covariance Estimate for Short-term Portfolio Optimization. *Journal of Machine Learning Research*, 21(97): 1–37.

Lai, Z.-R.; Wu, X.; Fang, L.; and Chen, Z. 2024. A Desingularity Subgradient Approach for the Extended Weber Location Problem. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*.

Lai, Z.-R.; and Yang, H. 2023. A Survey on Gaps between Mean-variance Approach and Exponential Growth Rate Approach for Portfolio Optimization. *ACM Computing Surveys*, 55(2): 1–36. Article No. 25.

Lai, Z.-R.; Yang, P.-Y.; Fang, L.; and Wu, X. 2018c. Short-term Sparse Portfolio Optimization Based on Alternating Direction Method of Multipliers. *Journal of Machine Learning Research*, 19(63): 1–28.

Li, B.; Hoi, S. C. H.; Zhao, P.; and Gopalkrishnan, V. 2013. Confidence Weighted Mean Reversion Strategy for Online Portfolio Selection. *ACM Transactions on Knowledge Discovery from Data*, 7(1). Article 4.

Li, B.; Sahoo, D.; and Hoi, S. C. 2016. OLPS: a toolbox for on-line portfolio selection. *Journal of Machine Learning Research*, 17(1): 1242–1246.

Lin, Y.; Zhang, Y.; Lai, Z.-R.; and Li, C. 2024. Autonomous Sparse Mean-CVaR Portfolio Optimization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 30440–30456. PMLR.

Morris, J. G. 1981. Convergence of the Weiszfeld algorithm for Weber problems using a generalized "distance" function. *Operations Research*, 29(1): 37–48.

Ostresh, L. M. 1978. On the convergence of a class of iterative methods for solving the weber location problem. *Operations Research*, 26(4): 597–609.

Rockafellar, R. T.; and Wets, R. J. B. 2009. *Variational Analysis*, volume 317. Springer Science & Business Media.

Sharpe, W. F. 1966. Mutual Fund Performance. *Journal of Business*, 39(1): 119–138.

Vardi, Y.; and Zhang, C.-H. 2000. The Multivariate ℓ^1-Median and Associated Data Depth. *Proceedings of the National Academy of Sciences of the United States of America*, 97(4): 1423–1426.

Weber, A. 1909. *Uber den Standort der Industrien*. Tubingen : Mohr.