2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

An Efficient Two-Step Approach to Fair and Sparse Transactions Allocation

Yinjie He, Jian Guo, Ruoshi Shi and Yanlong Zhao

Abstract— This study considers a practically important financial transactions allocation problem originated from interbank market. To achieve fairness and sparsity at the same time, we formulate the problem as a non-linear sparse optimization problem. A novel two-step algorithm that (1) finds the most sparse but not necessarily fair solution, (2) then utilizes iterative local adjustments to cope with non-linear fairness constraint is proposed. An adaptive parameter selection method to improve efficiency, avoiding time-consuming parameter search is devised. We provide theoretical guarantees that the two-step algorithm along with the adaptive parameter selection can always find a feasible solution with as much sparsity as possible. The effectiveness and efficiency of the algorithm is demonstrated by conducting empirical analysis on a real dataset from financial industry.

Index Terms— Interbank lending, nonlinear optimization, regularization algorithm, sparse identification, transactions allocation

I. Introduction

Interbank lending market facilitates short-term borrowing and lending transactions among financial institutions, which contributes to the stability of the financial system [1] [2]. Typically, asset management companies conduct such transactions by allocating a portion of the capital from various portfolios which lend money, hereafter referred to as lenders. Each lender has different lending amounts [3]. And different borrowers have different demands, vary in amount and the interest rate [4]. It is challenging for traders to align the lenders with the borrowers manually, which leads to the transaction allocation problem.

Manual transaction allocation faces operational risks [5]. As the number of lenders and borrowers [6] increases, manual allocation can be time-consuming, and prone to errors. Furthermore, it may be subject to individual biases and lack of fairness [7]. It is necessary to develop a transactions allocation algorithm, which addresses three main concerns: fairness, sparsity and efficiency.

Fairness. We utilize the difference between the highest and lowest values of the rates obtained by lenders as a metric of fairness. In practice, trading department usually demands that the difference should be less than a given threshold.

Sparsity. Driven by regulatory compliance and transaction costs [8], trading departments also demand that each borrower's transactions should ideally not be excessively fragmented, which imposes sparsity requirements.

Efficiency. Efficiency is dictated by the immediacy of the financial problems. The trading department cannot afford excessively lengthy solving time.

There are two main challenges. The first is that transaction allocation which requires fairness and sparsity is rarely seen in academic field. The other challenge is that the algorithm is required to adapt to problems of different scales , i.e., different number of lenders and borrowers or different amounts of trading volume.

In this study, we seek to formulate the transactions allocation problem as a sparse nonlinear optimization problem. A novel two-step algorithm with adaptive parameters search are proposed, with a theoretical guarantee of convergence to a sparse feasible solution that meets fairness requirements even in high-dimensional situations. The main contributions are as follows:

1) We present a novel two-step algorithm, which (1) finds the most sparse but not necessarily fair solution, (2) and iteratively adjusts the allocation locally until the fairness constraint is satisfied, circumventing the challenges of global adjustment, to solve the transactions allocation problem.
2) We devise adaptive parameter selection for the locally adjusting step of the algorithm, which improves the efficiency of the algorithm. We provide theoretical justifications that the two-step algorithm along with adaptive parameter selection can be guaranteed to find a sparse solution that satisfies the fairness constraints.
3) We demonstrate our two-step approach by experiments based on a real world dataset. The experimental results show that the algorithm meets the theoretical expectations and surpasses traditional methods in terms of sparsity and solution time.

The remainder of the paper is organized as follows. Section II formulates the problem. Section III reviews the literature pertinent to solving the above problem. Section IV presents the algorithm and provides theoretical guarantees. Section V demonstrates the effectiveness of the two-step approach.

The work is supported by National Natural Science Foundation of China under Grants 62025306. (Corresponding author: Yanlong Zhao.)

Yinjie He, Jian Guo, and Yanlong Zhao are with the Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, and School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, P. R. China. Emails: heyinjie@amss.ac.cn, j.guo@amss.ac.cn, ylzhao@amss.ac.cn

R.S. Shi is with the Postdoctoral Research Workstation of Industrial and Commercial Bank of China, 100032 China, on leave from of Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100140 China (e-mail: shiruoshi18@amss.ac.cn).

979-8-3503-1632-2/24/$31.00 ©2024 IEEE

II. Problem Formulation

Notation: Without loss of generality, we consider m lenders and n borrowers. Let $\mathbf{l} = (l_1, l_2, \ldots, l_m)^T$ with $l_i \geq 0$ for $i = 1, \ldots, m$ denote the available lending amount of each lender and let $\mathbf{b} = (b_1, b_2, \ldots, b_n)^T$ with $b_j \geq 0$ for $j = 1, \ldots, n$ denote the borrowing amount of each borrower. Denote $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$ as the interest rates offered by each borrower. We can always assume that $\sum_{i=1}^{m} l_i = \sum_{j=1}^{n} b_j$, which means that total borrowing amount equals total lending amount as it corresponds to real-world scenarios. Let the matrix

$$\mathbf{H} = \left(h_{ij}\right)_{m \times n}$$

denote the allocation. In the allocation matrix $\mathbf{H}$, the element h_{ij} denotes the transaction volume between the i_{th} lender and the j_{th} borrower. For a column vector $\mathbf{x} = (x_1, x_2, \ldots, x_k)^T$, let $\max(\mathbf{x})$ denote the maximum element of $\mathbf{x}$ and $\min(\mathbf{x})$ denote the minimum element. We then introduce the definition of the weighted interest rate $\mathbf{i}$ of lenders under $\mathbf{H}, \mathbf{i} = \mathbf{Hr} \oslash \mathbf{l}$.

Remark 1. In the above equation, $\oslash$ signifies element-wise division. And $\mathbf{i} = (i_1, i_2, \ldots, i_m)^T$ is an m-dimensional column vector, where each element represents the weighted interest rate of the lender.

Modelling. To ensure fairness, the difference between the highest and lowest values of $\mathbf{i}$ should be less than a given fair threshold e, i.e., $\max(\mathbf{i}) - \min(\mathbf{i}) \leq e$, where $\mathbf{i} = \mathbf{Hr} \oslash \mathbf{l}$. Note that trade volume should be non-negative, so for allocation matrix $\mathbf{H}$, each element h_{ij} should be non-negative, i.e., $h_{ij} \geq 0$ for $i = 1, \ldots, m, j = 1, \ldots, n$. In addition, we should ensure that the total trading volume of a lender equals its lending amount, and that the total trading volume of a borrower equals its borrowing amount. Let $\mathbf{1}_m = (1, 1, \ldots, 1)^T$, then we have $\mathbf{H1}_n = \mathbf{l}, \quad \mathbf{1}_m^T \mathbf{H} = \mathbf{b}^T$.

Furthermore, as discussed before, we should not overly fragment the transactions with a single borrower. This means that we need to minimize $\|\mathbf{H}\|_0$, where $\|\mathbf{H}\|_0$ refers to the L_0 norm of $\mathbf{H}$. Formally, the sparse transaction allocation problem can be described as

$$\begin{aligned}
\min_{\mathbf{H}} \quad & \|\mathbf{H}\|_0 && (1)\\
\text{s.t.} \quad & \max(\mathbf{i}) - \min(\mathbf{i}) \leq e, && (2)\\
& \mathbf{H1}_n - \mathbf{l} = 0, && (3)\\
& \mathbf{1}_m^T \mathbf{H} - \mathbf{b}^T = 0, && (4)\\
& h_{ij} \geq 0, \forall i, j. && (5)
\end{aligned}$$

Goal. Our goal is to make the allocation matrix $\mathbf{H}$ as sparse as possible while ensuring fairness and satisfying other constraints.

III. Related Work

This section presents some related work(1). Firstly, the L_0 norm, or alternatively, the L_0-penalty actually yields the most sparse solutions. However, minimizing L_0-penalty involves addressing an NP hard optimization [9], rendering it infeasible for situations requiring efficiency. Consequently, we employ regularization methods as the approach for solving transaction allocation. Generally, we replace $\|\mathbf{H}\|_0$ with some certain sparse penalty function $P(\mathbf{H})$,

$$\min_{\mathbf{H}} \quad P(\mathbf{H}), \tag{6}$$

where we omit the constraints for simplicity.

Regarding sparse penalty functions, a common alternative to the L_0-penalty is the L_1-penalty [10]. But we cannot apply L_1-penalty here because each element in $\mathbf{H}$ is non-negative and the sum of each row and column is controlled by the constraints, which means that $\|\mathbf{H}\|_1$ always equals to a certain constant. Fan and Li [11] proposed Smoothly Clipped Absolute Deviation (SCAD) penalty function which reduces penalty on significant elements. But we have to set 2 parameters for SCAD carefully to effectively utilize it. Chartrand [12] proposed that L_1-penalty can be replaced by L_p-penalty with $0 < p < 1$. More specifically, Xu [13] pointed out that by setting p to $1/2$, a balance between efficiency and sparsity can be achieved. The choice of sparse penalty functions will be detailed later.

In general, assume that we have already picked a penalty function, we should consider how to solve the constrained problem. It is relatively easy to handle the linear constraints (3)-(5). On satisfying the highly non-linear fairness constraint (2), relaxation methods are usually implemented. A common approach is complicated linear programming relaxation. But it has very high computation complexity in large-scale problem [14]. Another alternative is also to relax the fairness constraint but to minimize the variance of the weighted interest rate of lenders. Minimizing the variance might lead to a serious violation of fairness constraint because it cannot ensure that the difference between the highest and lowest values of weighted interest rate of lenders will satisfy the constraint even when the minimum is achieved.

Given this non-linear sparse optimization problem, straightforwardly implementing relaxations and seeking an optimal solution requires excessive computational resources or may even fail in limited time. In the next section, we will propose a provably good two-step approach to efficiently solve the problem.

IV. Two-Step Approach

In this section, we present a novel provably good two-step algorithm to address this problem. Intuitively, in the first step, we ignore the fairness constraint and relax the problem to a sparse optimization problem(6) only with the linear constraints (3)-(5), which is relatively easy to solve. In the second step, to ensure that the solution obtained in the first step satisfies the fairness constraint, we iteratively select two lenders with the highest and lowest weighted interest rates, making slight adjustments

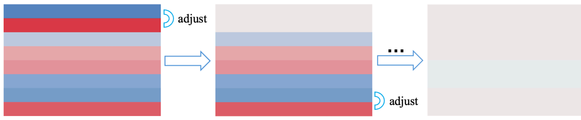

Fig. 1. This figure is a demonstration of the refining step on the allocation matrix. Each horizontal line represents a lender, with red indicating that the lender's weighted interest rate is higher than the fair value, and blue indicating the opposite. The deeper the color, the greater the extent. Iterative local adjustments between the highest rate lender and the lowest value lender ensure fairness.

to narrow the gap between their rates and maintain the sparsity as much as possible.

A. The most sparse solution under relaxation of fairness

Given the amounts $\mathbf{l}$ and $\mathbf{b}$, and the interest rates offered by borrowers $\mathbf{r}$, we formulate the problem:

$$\begin{aligned} \min_{\mathbf{H}} \quad & P(\mathbf{H}), \\ \text{s.t.} \quad & \max(\mathbf{i}) - \min(\mathbf{i}) \le e, \\ & \mathbf{H}\mathbf{1}_n - \mathbf{l} = 0, \\ & \mathbf{1}_m^T\mathbf{H} - \mathbf{b}^T = 0, \\ & h_{ij} \ge 0, \forall i, j. \end{aligned} \tag{7}$$

Here we do not specify the choice of a specific penalty function to maintain the generality of the algorithm.

As discussed before, we first circumvent the nonlinear fairness constraints and focus on the remaining linear constraints,

$$\begin{aligned} \min_{\mathbf{H}} \quad & P(\mathbf{H}), \\ \text{s.t.} \quad & \mathbf{H}\mathbf{1}_n - \mathbf{l} = 0, \\ & \mathbf{1}_m^T\mathbf{H} - \mathbf{b}^T = 0, \\ & h_{ij} \ge 0, \forall i, j. \end{aligned} \tag{8}$$

We can directly solve this simplified problem using a nonlinear optimization solver and let $\mathbf{H}_0$ denote the solution obtained in this step. In the first step, we relax the fairness constraint to search for the optimal sparse solution $\mathbf{H}_0$ within a broader scope. In fact, there is a chance that $\mathbf{H}_0$ luckily falls into the smaller area that satisfies the fairness constraint. In that case, $\mathbf{H}_0$ would also be the optimal solution to the original problem. Otherwise, if $\mathbf{H}_0$ violates the fairness constraint, we have to make additional adjustments to ensure fairness. The methods will be detailed below.

B. Refining fairness within constraints

If fairness constraint is not satisfied, we will adjust the allocation to ensure the fairness. To maintain efficiency, we prefer not to implement global changes on the allocation matrix. Alternatively, we employ iterative local adjustments to gradually meet the fairness constraint.

More specifically, at each iteration, we isolate the two lenders with the highest and lowest rates. After some proper adjustments, we re-evaluate the rates of all the lenders. If fairness is obtained, the loop concludes. Otherwise, we proceed to the next iteration.

First layer iteration. Suppose we get the result $\mathbf{H}_k$, $k \ge 0$ for the k_{th} iteration, $\mathbf{H}_0$ refers to the allocation obtained in the first step. At this layer we solve a subproblem to make the two lenders mentioned before satisfy fairness without changing other constraints. Assume that the a_{th} lender has the highest rate and the b_{th} lender has the lowest rate. Correspondingly, the allocation of the a_{th} lender for the k_{th} iteration is $(h_{a,1}^{k,0}, h_{a,2}^{k,0}, \ldots, h_{a,n}^{k,0})^T$ and the allocation of the other one is $(h_{b,1}^{k,0}, h_{b,2}^{k,0}, \ldots, h_{b,n}^{k,0})^T$. Now we sum these two vectors and formulate the subproblem to complete the adjustment. In this subproblem, we only have two lenders who have lending amount l_a and l_b, and n borrowers as before but with the borrowing amount summed by $\tilde{\mathbf{b}}_k^T = (h_{a,1}^{k,0}+h_{b,1}^{k,0}, h_{a,2}^{k,0}+h_{b,2}^{k,0}, \ldots, h_{a,n}^{k,0}+h_{b,n}^{k,0})^T$. The interest rate commited by each borrower is still $\mathbf{r}_k = (r_1, r_2, \ldots, r_n)^T$. Note that in this subproblem, we can relax the fairness constraint into a quadratic function with sparse penalty. Let $\tilde{\mathbf{H}}_k^0 = \begin{pmatrix} \tilde{\mathbf{h}}_{k,0}^1 \\ \tilde{\mathbf{h}}_{k,0}^2 \end{pmatrix} = \begin{pmatrix} h_{a,1}^{k,0}, h_{a,2}^{k,0}, \ldots, h_{a,n}^{k,0} \\ h_{b,1}^{k,0}, h_{b,2}^{k,0}, \ldots, h_{b,n}^{k,0} \end{pmatrix} \in \mathbb{R}^{2\times n}$ denote the initial allocation matrix with two lenders and n borrowers. Next, given the penalty parameter ρ_0 at the first iteration in the first layer, we solve the following subproblem to make these two lenders a and b satisfy the fairness requirement:

$$\begin{aligned} \min_{\tilde{\mathbf{H}}=(\tilde{h}_{i,j})\in\mathbb{R}^{2\times n}} \quad & (i_a - i_b)^2 + \rho_0 P(\tilde{\mathbf{H}}), \\ \text{s.t.} \quad & \tilde{\mathbf{H}}\mathbf{1_n} - \begin{pmatrix} l_a \\ l_b \end{pmatrix} = 0, \\ & \mathbf{1}_2^T\tilde{\mathbf{H}} - \tilde{\mathbf{b}}_k^T = 0, \\ & \tilde{h}_{ij} \ge 0, \forall i, j, \\ & \begin{pmatrix} i_a \\ i_b \end{pmatrix} = \tilde{\mathbf{H}}\mathbf{r} \oslash \begin{pmatrix} l_a \\ l_b \end{pmatrix}. \end{aligned} \tag{9}$$

The solution of the above subproblem is written as $\tilde{\mathbf{H}}_k^1$. We should determine ρ_0 carefully because it will influence both sparsity and fairness. For this ρ_0, we check whether the solution to the subproblem satisfies the fairness constraint (2) after the first iteration. If under the penalty coefficient ρ_0, the minimum point $\tilde{\mathbf{H}}_k^1$ of subproblem (9) fails to ensure the fairness, we can update the penalty coefficient ρ_0 by using a decay function $d(x)$ to update ρ_0 as $\rho_1 = d(\rho_0)$. We then continue with the first layer of iterations, solving the subproblem (9) and checking fairness under the new penalty coefficient ρ_1. After p iterations, we get a solution that satisfies fairness (2), denoted as

$$\tilde{\mathbf{H}}_k^p = \begin{pmatrix} \tilde{\mathbf{h}}_{k,p}^1 \\ \tilde{\mathbf{h}}_{k,p}^2 \end{pmatrix} \in \mathbb{R}^{2\times n}. \tag{10}$$

Second layer iteration. In this layer, we first replace the a_{th} and b_{th} rows of the original $\mathbf{H}_k$ with the two rows of (10) in the first layer to get the new matrix denoted as $\mathbf{H}_{k+1}$. We then use the resulting $\mathbf{H}_{k+1}$ to continue. A demonstration of the two-layer iteration can be seen in Figure 1. Continuing this two-layer iteration until there exists K such that the complete matrix $\mathbf{H}_K$ satisfies the fairness condition, we end the iteration and output the final solution as $\mathbf{H}_K$. The specific algorithmic flow is described in Algorithm 1.

It can be theoretically guaranteed that if we choose ρ_p in the manner described above, we can obtain a solution $\tilde{\mathbf{H}} = (\tilde{h}_{ij})_{2\times n}$ to the subproblem (9) that satisfies both fairness constraints and linear constraints while maintaining sparsity at each iteration. Let $\tilde{\mathbf{H}}_k^p$ denote the minimum value of the subproblem (9) under p_{th} first layer iteration and k_{th} second layer iteration determined by the penalty coefficient $\rho_p = d^p(\rho_0)$. Here, $d^p(\cdot)$ denotes a p_{th} composite of the function $d(\cdot)$.

Prop 1. Given the fairness constraint $\max(\mathbf{i}) - \min(\mathbf{i}) \leq e$. If d satisfies $d^p(x) \to 0$ as $p \to \infty$ for any $x > 0$, then there exists a positive integer q such that $|i_a - i_b| \leq e$, where $\begin{pmatrix} i_a \\ i_b \end{pmatrix} = \tilde{\mathbf{H}}_k^q \mathbf{r} \oslash \begin{pmatrix} l_a \\ l_b \end{pmatrix}$.

Proof. If $p \to \infty$, the sparse penalty coefficient $\rho_p = d^p(\rho_0) \to 0$. It is obvious that there exists an optimal solution to (9) with $\rho_0 = 0$. We denote $\tilde{\mathbf{H}}_k^*$ as it. And we let $J_p(\tilde{\mathbf{H}}) = (i_a - i_b)^2 + \rho_p P(\tilde{\mathbf{H}})$, where $\rho_p = d^p(\rho_0)$. Note that $(i_a - i_b)^2$ is actually a quadratic function of $\tilde{\mathbf{H}}$, we let $Q(\tilde{\mathbf{H}}) = (i_a - i_b)^2(\tilde{\mathbf{H}})$. Without loss of generality, we assume that the penalty function $P(\tilde{\mathbf{H}})$ is always non-negtive and bounded since the elements of $\tilde{\mathbf{H}}$ should be bounded in feasible region according to the linear constraints. Therefore, all we have to prove is that there exists a positive integer q, such that $Q(\tilde{\mathbf{H}}_k^q) = (i_a - i_b)^2(\tilde{\mathbf{H}}_k^q) \leq e^2$. Note that $J_p(\tilde{\mathbf{H}}_k^p) \leq J_p(\tilde{\mathbf{H}}_k^*)$, so we have

$$\rho_p P(\tilde{\mathbf{H}}_k^p) + Q(\tilde{\mathbf{H}}_k^p) \leq \rho_p P(\tilde{\mathbf{H}}_k^*) + Q(\tilde{\mathbf{H}}_k^*). \tag{11}$$

On the other hand, since $P(\tilde{\mathbf{H}}_k^p)$ is non-negative, we have

$$\rho_p P(\tilde{\mathbf{H}}_k^p) + Q(\tilde{\mathbf{H}}_k^p) \geq Q(\tilde{\mathbf{H}}_k^p) \geq Q(\tilde{\mathbf{H}}_k^*). \tag{12}$$

Combining (11) and (12), we can deduce that

$$Q(\tilde{\mathbf{H}}_k^*) \leq \rho_p P(\tilde{\mathbf{H}}_k^p) + Q(\tilde{\mathbf{H}}_k^p) \leq \rho_p P(\tilde{\mathbf{H}}_k^*) + Q(\tilde{\mathbf{H}}_k^*).$$

Note that $P(\tilde{\mathbf{H}}_k^*)$ is a constant when p changes and $Q(\tilde{\mathbf{H}}_k^*) = 0$, so if we let $p \to \infty$, we obtain

$$0 \leq lim_{p\to\infty} Q(\tilde{\mathbf{H}}_k^p) \leq 0. \tag{13}$$

(13) implies the result. □

Another issue worth mentioning is that all the constraints included in (9), i.e., (3)-(5), will not be violated by $\mathbf{H}_{k+1}$ after k_{th} second iteration. This means that the refining step will not compromise the constraints. The result is given in the following proposition.

Prop 2. After k_{th} second layer iteration, the allocation matrix $\mathbf{H}_{k+1}$ incorporated by the solution to the subproblem (9) will satisfy the linear constraints (3)-(5).

Algorithm 1: Two Step Transactions Allocation Algorithm

Data: Lenders $\mathbf{l}$, borrowers $\mathbf{b}$, weighted interest rate $\mathbf{r}$, and threshold value e

Input: Initial penalty coefficient ρ_0, penalty function $P(\cdot)$, and decay function $d(\cdot)$. Set $k = 0$ and $p = 0$.

Output: Final allocation matrix H_K

. Step 1. Seek the most sparse solution under relaxation of fairness. Solve the following relaxation problem:

$$\begin{aligned} \min_{\mathbf{H}} \quad & P(\mathbf{H}), \\ \text{s.t.} \quad & \mathbf{H}\mathbf{1}_n - \mathbf{l} = 0, \\ & \mathbf{1}_m^T \mathbf{H} - \mathbf{b}^T = 0, \\ & h_{ij} \geq 0, \forall i, j. \end{aligned}$$

Then, get the initial estimate H_k and calculate $\mathbf{i}_k = \mathbf{H}_k \mathbf{r} \oslash \mathbf{l}$.

while $\max(\mathbf{i}_k) - \min(\mathbf{i}_k) > e$ do Corresponding to the second layer iteration, the two rows with the largest interest rate difference are adjusted to satisfy the fairness condition.

  Step 2. Refine the fairness within constraints. For $\mathbf{H}_k$, find the two rows with the largest difference in interest rates, noted as rows a and b and calculate $\tilde{\mathbf{b}}_{k,p}^T = (h_{a,1}^{k,p} + h_{b,1}^{k,p}, h_{a,2}^{k,p} + h_{b,2}^{k,p}, \ldots, h_{a,n}^{k,p} + h_{b,n}^{k,p})^T$ and define

$$\tilde{\mathbf{H}}_k^p = \begin{pmatrix} \tilde{\mathbf{h}}_{k,p}^1 \\ \tilde{\mathbf{h}}_{k,p}^2 \end{pmatrix} = \begin{pmatrix} h_{a,1}^{k,p}, h_{a,2}^{k,p}, \ldots, h_{a,n}^{k,p} \\ h_{b,1}^{k,p}, h_{b,2}^{k,p}, \ldots, h_{b,n}^{k,p} \end{pmatrix} \in \mathbb{R}^{2\times n}.$$

  while $\tilde{\mathbf{H}}_k^p$ does not satisfy the fairness condition for e. do Corresponding to the first layer iteration, the transaction allocation is adjusted by solving a subproblem so that it ultimately satisfies the fairness condition.

    Solve the following sub-problem:

$$\begin{aligned} \min_{\tilde{\mathbf{H}} = (\tilde{h}_{i,j}) \in \mathbb{R}^{2\times n}} \quad & (i_a - i_b)^2 + \rho_p P(\tilde{\mathbf{H}}), \\ \text{s.t.} \quad & \tilde{\mathbf{H}}\mathbf{1_n} - \begin{pmatrix} l_a \\ l_b \end{pmatrix} = 0, \\ & \mathbf{1}_2^T \tilde{\mathbf{H}} - \tilde{\mathbf{b}}_k^T = 0, \\ & \tilde{h}_{ij} \geq 0, \forall i, j, \\ & \begin{pmatrix} i_a \\ i_b \end{pmatrix} = \tilde{\mathbf{H}} \mathbf{r} \oslash \begin{pmatrix} l_a \\ l_b \end{pmatrix}. \end{aligned}$$

    $p = p + 1$;

    $\rho_p = d(\rho_{p-1})$.

  end

  Get the output $\tilde{\mathbf{H}}_k^P$. Replace the two rows in $\mathbf{H}_k$ that have the largest difference in interest rates with $\tilde{\mathbf{H}}_k^P$.

  $k = k + 1$.

end

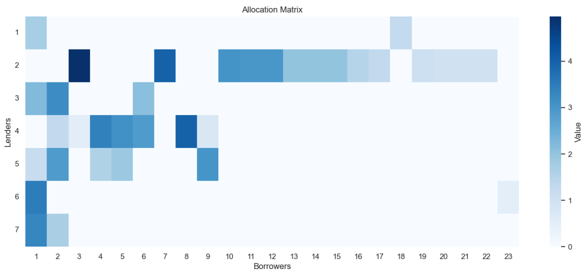

Fig. 2. Allocation matrix given by our two-step algorithm, with 7 lenders and 23 borrowers.

Proof. We proceeed by induction. Note that $\mathbf{H}_0$ obtained before satisfies all the linear constraints (3)-(5).

Now we assume that $\mathbf{H}_k$ satisfies (3)-(5). Given that our solution to (9) satisfies the local linear constraints stated in (9), so putting it back into the global allocation matrix will not violate the linear constraints (3)-(5). This implies that $\mathbf{H}_{k+1}$ satisfies (3)-(5). □

In summary, with the propositions above, we can reach our main result.

Theorem 1. Our two-step algorithm can always obtain an sparse solution while satisfying all the constraints including the fairness constraint.

Proof. We only have to consider the situation that $\mathbf{H}_0$ is not fair enough. In that case, we locally adjust the allocation matrix as is discussed above. By prop 1 and prop 2, we conclude the proof. □

Remark 2. In the theoretical justifications above, we do not specify penalty functions, which means that the effectiveness of the algorithm will not affected by the selection of penalty functions. However, we recommend using a penalty function with fewer parameters or even without parameters to enhance robustness and improve efficiency.

In the next section, we will conduct experiments based on real world dataset to evaluate the performance of the algorithm.

V. Empirical Analysis

In this section, we perform the allocation algorithm on a real dataset from industry and compare it with some baseline algorithms. We first introduce the data.

A. Dataset

Specifically, a company from industry provided a transactions allocation dataset which are originated from real data. Each problem contained in the dataset is determined by three columns of data, which are lending amounts, borrowing amounts, and corresponding interest rates. Note that the numbers of lenders and borrowers in each problem differ. The largest problem involves 7 lenders and 23 borrowers while the smallest problem involves 2 lenders and 5 borrowers. We will present the smallest data sample, where the units of amount are in hundreds of millions of RMB.

lender amount $= (6, 15)^T$,

borrower amount $= (3.04, 1.96, 10, 2.2, 3.8)^T$,

borrower rate $= (1.9\%, 1.91\%, 1.91\%, 1.94\%, 1.94\%)^T$.

Since transactions allocation is a deterministic optimization problem, we do not necessarily need a huge amount of data. The dataset contains eight samples. In fact, the problems contained in the dataset cover typical scenarios that a trader would encounter in real world.

B. Experiment Design

Baseline. We first introduce the benchmark algorithms utilized in our experiment. Quadratic relaxation is used to transform the original problem(7) into a quadratic sparse optimization problem with linear constraints. We can formulate it as

$$\begin{aligned} \min_{\mathbf{H}} \quad & \rho P(\mathbf{H}) + \gamma \sum_{p,q} (i_p - i_q)^2, \\ \text{s.t.} \quad & \mathbf{H}\mathbf{1}_n - \mathbf{l} = 0, \\ & \mathbf{1}_m^T \mathbf{H} - \mathbf{b}^T = 0, \\ & h_{ij} \geq 0, \forall i, j. \end{aligned} \tag{14}$$

where ρ is sparse penalty coefficient, $\mathbf{i} = \mathbf{Hr} \oslash \mathbf{l}$,and $\mathbf{i} = (i_1, i_2, \ldots, i_m)^T$.

On the selection of sparse penalty function, we consider $L_{1/2}$-penalty and SCAD penalty as alternatives because the choice of parameters within them is relatively robust. In fact, $L_{1/2}$-penalty does not require any parameters. Formally, if we apply $L_{1/2}$-penalty, then $P(\mathbf{H}) = P_{L_{1/2}}(\mathbf{H}) = \sum_{i,j}(h_{ij}^{1/2})$. And if we apply SCAD penalty, then $P(\mathbf{H}) = P_{SCAD}(\mathbf{H}) = \sum_{i,j} P_\lambda(h_{ij})$, where

$$P_\lambda(h_{ij}) = \begin{cases} \lambda|h_{ij}| & \text{if } |h_{ij}| \leq \lambda, \\ \frac{-h_{ij}^2 + 2a\lambda|h_{ij}| - \lambda^2}{2(a-1)} & \text{if } \lambda < |h_{ij}| \leq a\lambda, \\ \frac{(a+1)\lambda^2}{2} & \text{if } |h_{ij}| > a\lambda, \end{cases}$$

where $\lambda > 0$ is the regularization parameter and $a > 2$ is a constant that controls the concavity. We always fix $a = 3.7$ as recommended in [11].

Note that in both cases we have to select coefficients ρ and γ in (14) depending on the particular problems. Besides, we also have to determine λ in SCAD penalty. Therefore, to maximize the likelihood of successfully find a fair solution, we utilize grid search when solving (14).

Two-step Algorithm. In the experiment, we have to specify some necessary details. We first explain the input of our two-step algorithm. We set $\rho_0 = 1$, set decay function $d^n(x) = s^n x$ with $s = 0.1$. On the penalty function, we utilize $L_{1/2}$-penalty as it does not require additional parameters. As is widely adopted in the industry, we should set the fairness threshold $e = 0.02\%$.

General Setting. The non-linear optimization problems are solved using the SciPy library within the Python 3.11 environment. The device utilized is an Apple MBP14 with the M1 Pro processor.

C. Results

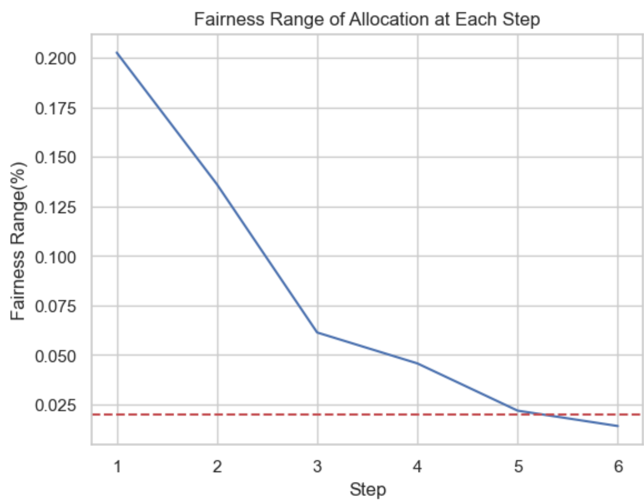

Fig. 3. The blue line displays the changing process of the difference between the highest and lowest rate. X-axis refers to the number of second layer iterations. The red dashed line represents the threshold value $e = 0.02\%$, which is widely adopted in financial industry. Once we manage to find an allocation with a fairness range below the threshold, we successfully find a solution.

The experiment results are shown in Table I. 'Fairness Ratio' refers to the proportion of successfully finding a fair solution, which means that the fairness range of the allocation generated by the algorithm is below the threshold. 'Sparsity' refers to the average proportion of non-zero elements of the allocation matrix. 'Average Time' refers to the average processing time. We can observe that the two-step algorithm always satisfies fairness constraint with 100% success. In addition, the proposed algorithm has the lowest average non-zero ratio, which implies that it provides more sparse allocation on average than the baseline algorithms. Notably, our two-step algorithm only takes 31.62s to provide a satisfying allocation, much faster than the baseline. The algorithm takes 11.8% of the time compared to Quad-$L_{1/2}$ and takes 6.7% of the time compared to Quad-SCAD.

TABLE I
This table displays the results.

Algorithm	Fairness Ratio	Sparsity	Average Time
Two-Step	100%	46.87%	31.62s
Quad-SCAD	62.5%	50.15%	469.42s
Quad-$L_{1/2}$	62.5%	56.88%	267.69s

Figure 3 shows the influence of the iterative adjustments on the interest rate range among the lenders. We can observe that fairness is gradually obtained step by step, as is expected theoretically. In addition, we present a fair allocation scheme obtained in one problem in Figure 2. It is quite straightforward to perceive the sparsity of the solution given by the proposed algorithm.

VI. Conclusions

Motivated by the practically important interbank lending transaction allocation problem, we consider a non-linear sparse optimization problem with complicated constraints. To ensure fairness, sparsity, and seek more efficient solution, we develop a novel two-step algorithm which can be implemented even in high-dimensional situations. It employs iterative local adjustments to ensure compliance with highly non-linear fairness constraints. We also devise an adaptive parameters selection method, making the algorithm generalize across problems of varying scales without changing the initial parameters. In addition, we lay the theoretical foundation for our two-step approach. We provide a proof that the two-step algorithm along with the adaptive parameter selection method always provide a sparse feasible solution. On the other hand, we present empirical analysis on a real world dataset in Python. The results across different scale problems show that the two-step algorithm is capable of ensuring fairness and sparsity, while significantly improving speed.

References

[1] Jean-Charles Rochet and Jean Tirole. Interbank lending and systemic risk. Journal of Money, credit and Banking, 28(4):733–762, 1996.

[2] Falk Bräuning and Falko Fecht. Relationship lending in the interbank market and the price of liquidity. Review of Finance, 21(1):33–75, 2017.

[3] Selva Demiralp, Brian Preslopsky, and William Whitesell. Overnight interbank loan markets. Journal of Economics and Business, 58(1):67–83, 2006.

[4] Damir Filipović and Anders B Trolle. The term structure of interbank risk. Journal of Financial Economics, 109(3):707–733, 2013.

[5] Robert A Jarrow. Operational risk. Journal of Banking & Finance, 32(5):870–879, 2008.

[6] Roland Gillet, Georges Hübner, and Severine Plunus. Operational risk and reputation in the financial industry. Journal of Banking & Finance, 34(1):224–235, 2010.

[7] Allen N Berger, Filippo Curti, Atanas Mihov, and John Sedunov. Operational risk is more systemic than you think: Evidence from us bank holding companies. Journal of Banking & Finance, 143:106619, 2022.

[8] Jens Dick-Nielsen and Marco Rossi. The cost of immediacy for corporate bonds. The Review of Financial Studies, 32(1):1–41, 2019.

[9] Zhanxuan Hu, Feiping Nie, Rong Wang, and Xuelong Li. Low rank regularization: A review. Neural Networks, 136:218–232, 2021.

[10] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1):267–288, 1996.

[11] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association, 96(456):1348–1360, 2001.

[12] Rick Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. IEEE Signal Processing Letters, 14(10):707–710, 2007.

[13] Zongben Xu, Hai Zhang, Yao Wang, XiangYu Chang, and Yong Liang. L 1/2 regularization. Science China Information Sciences, 53:1159–1169, 2010.

[14] Yunpeng Li, Changjie He, Yichuan Jiang, Weiwei Wu, Jiuchuan Jiang, Wei Zhang, and Hui Fan. Max-min fair allocation for resources with hybrid divisibilities. Expert Systems with Applications, 124:325–340, 2019.