Statistics and Computing

Optimal non-negative forecast reconciliation

--Manuscript Draft--

Manuscript Number:	
Full Title:	Optimal non-negative forecast reconciliation
Article Type:	Manuscript
Keywords:	Hierarchical time series; coherent forecasts; reconciliation; non-negative least squares; algorithms; forecast combination; aggregation of data; fast algorithms
Corresponding Author:	Shanika Wickramasuriya, Ph.D University of Auckland Auckland, NEW ZEALAND
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of Auckland
Corresponding Author's Secondary Institution:	
First Author:	Shanika Wickramasuriya, Ph.D.
First Author Secondary Information:	
Order of Authors:	Shanika Wickramasuriya, Ph.D.
	Berwin Turlach
	Rob Hyndman
Order of Authors Secondary Information:	
Funding Information:	

Statistics and Computing manuscript No.
(will be inserted by the editor)

Optimal non-negative forecast reconciliation

Shanika L Wickramasuriya · Berwin A Turlach · Rob J Hyndman

Received: date / Accepted: date

Abstract The sum of forecasts of a disaggregated time series are often required to equal the forecast of the aggregate. The least squares solution for finding coherent forecasts uses a reconciliation approach known as MinT, proposed by Wickramasuriya, Athanasopoulos, and Hyndman (2019). The MinT approach and its variants do not guarantee that the coherent forecasts are non-negative, even when all of the original forecasts are non-negative in nature. This has become a serious issue in applications that are inherently non-negative such as with sales data or tourism numbers. While overcoming this difficulty, we consider the analytical solution of MinT as a least squares minimization problem. The non-negativity constraints are then imposed on the minimization problem to ensure that the coherent forecasts are strictly non-negative.

Considering the dimension and sparsity of the matrices involved, and the alternative representation of MinT, this constrained quadratic programming problem is solved using three algorithms. They are the block principal pivoting algorithm, projected conjugate gradient algorithm, and scaled gradient projection algorithm. A Monte Carlo simulation is performed to evaluate the computational performances of these algorithms. The results demonstrate that the block principal pivoting algorithm clearly outperforms the rest, and projected conjugate gradient is the second best. The superior performance of the block principal pivoting algorithm can be partially attributed to the alternative representation of the weight matrix in the MinT approach.

An empirical investigation is carried out to assess the impact of imposing non-negativity constraints on forecast reconciliation. It is observed that slight gains in forecast accuracy have occurred at the most disaggregated level. At the aggregated level slight losses are also observed. Although the gains or losses are negligible, the procedure plays an important role in decision and policy implementation processes.

Keywords Aggregation · Coherent forecasts · Forecast reconciliation · Hierarchical · Least squares · Non-negative.

Shanika L Wickramasuriya (Corresponding author)
Department of Statistics, University of Auckland, Auckland, New Zealand
ORCID: 0000-0003-2742-5992
E-mail: s.wickramasuriya@auckland.ac.nz

Berwin A Turlach
Centre for Applied Statistics, The University of Western Australia, Crawley, Australia
ORCID: 0000-0001-8795-471X
E-mail: berwin.turlach@uwa.edu.ac

Rob J Hyndman
Department of Econometrics and Business Statistics, Monash University, Australia
ORCID: 0000-0002-2140-5352
E-mail: rob.hyndman@monash.edu

1 Introduction

Forecast reconciliation is the problem of ensuring that disaggregated forecasts add up to the corresponding forecasts of the aggregated time series. This is a common problem in manufacturing, for example, where time series of sales are disaggregated in several ways — by region, product-type, and so on. There are often tens of thousands of forecasts at the most disaggregated level, and these are required to sum to give forecasts at higher levels of aggregation — a property known as "coherence".

A simple solution would be to forecast the most disaggregated series and sum the results. However, this

tends to give poor aggregate forecasts due to the low signal-to-noise ratio in the disaggregated time series. Instead, a better solution is to forecast all the series at all levels of aggregation, and then reconcile them so they are coherent; that is, so that the forecasts of the disaggregated series add up to the forecasts of the aggregated series. Least squares reconciliation was proposed by Hyndman et al. (2011), whereby the reconciled forecasts are as close as possible (in the L_2 sense) to the original ("base") forecasts subject to the aggregation constraint. This result was extended by Hyndman, Lee, and Wang (2016) to a larger class of reconciliation problems, and by Wickramasuriya, Athanasopoulos, and Hyndman (2019) who showed that the resulting reconciled forecasts are minimum variance unbiased estimators.

Most of the applications that we have found are inherently non-negative in nature, where the time series take only non-negative values such as revenue, demand, and counts of people. In such circumstances, it is important to ensure that the reconciled forecasts are non-negative, as forecasters and practitioners need to be able to make meaningful managerial decisions. Unfortunately, the MinT approach proposed by Wickramasuriya, Athanasopoulos, and Hyndman (2019) and its variants fail to guarantee this property even when the base forecasts are non-negative.

This can be overcome by explicitly imposing the non-negativity constraints on the reconciliation procedure. One simple technique is to overwrite any negatives in the reconciled forecasts with zeros. However, the resulting approximate solution has ill-defined mathematical properties. This type of overwriting approximation method is used in alternating least squares estimations (Berry et al., 2007; Karjalainen and Karjalainen, 1991). The problem is that it does not necessarily lower the objective function in each iteration, and therefore the convergence of the solution to the least squares minimum is not guaranteed. There are available algorithms that solve these problems in a mathematically rigorous way; we discuss how these can be applied to the forecast reconciliation problem.

2 The optimization problem

2.1 Notation and MinT reconciliation

We follow the notation of Wickramasuriya, Athanasopoulos, and Hyndman (2019) and let $\boldsymbol{y}_t \in \mathbb{R}^m$ denote a vector of observations at time t comprising all series of interest including both disaggregated and aggregated time series. We also define $\boldsymbol{b}_t \in \mathbb{R}^n$ to be the vector of the most disaggregated series at time t. These two vectors are connected via $\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{b}_t$ where $\boldsymbol{S}$ is the "summing" matrix of order $m \times n$ showing how the various aggregated time series in $\boldsymbol{y}_t$ are constructed from the disaggregated series in $\boldsymbol{b}_t$.

Let $\hat{\boldsymbol{y}}_T(h)$ is a vector of original (base) h-step-ahead forecasts, made at time T, stacked in the same order as $\boldsymbol{y}_t$. These will not generally be coherent. The least squares reconciled forecasts are given by

$$\tilde{\boldsymbol{y}}_T(h) = \boldsymbol{S}\tilde{\boldsymbol{b}}_T(h),$$

where

$$\tilde{\boldsymbol{b}}_T(h) = \left[(\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1}\boldsymbol{S})^{-1}\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1}\right] \hat{\boldsymbol{y}}_T(h), \qquad (1)$$

and $\boldsymbol{\Lambda}_h$ is a weighting matrix. Wickramasuriya, Athanasopoulos, and Hyndman (2019) showed that setting $\boldsymbol{\Lambda}_h = \text{var}[\boldsymbol{y}_{t+h} - \hat{\boldsymbol{y}}_t(h) \mid \boldsymbol{\mathcal{I}}_t]\}$ where $\boldsymbol{\mathcal{I}}_t = \{\boldsymbol{y}_t, \boldsymbol{y}_{t-1}, \dots\}$ to be the covariance matrix of the h-step-ahead base forecast errors minimizes the trace of $\text{var}[\boldsymbol{y}_{t+h} - \tilde{\boldsymbol{y}}_t(h) \mid \boldsymbol{\mathcal{I}}_t]$ amongst all possible unbiased reconciliations. They derived an alternative expression for $\tilde{\boldsymbol{b}}_T(h)$:

$$\tilde{\boldsymbol{b}}_T(h) = \left[\boldsymbol{J} - \boldsymbol{J}\boldsymbol{\Lambda}_h\boldsymbol{U}(\boldsymbol{U}^\top\boldsymbol{\Lambda}_h\boldsymbol{U})^{-1}\boldsymbol{U}^\top\right] \hat{\boldsymbol{y}}_T(h), \qquad (2)$$

where $\boldsymbol{J} = \left[\, \boldsymbol{0}_{n\times(m-n)} \,\vdots\, \boldsymbol{I}_n \,\right]$, $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{C}_{(m-n)\times n} \\ \hdashline \boldsymbol{I}_n \end{bmatrix}$, and $\boldsymbol{U}^\top = \left[\, \boldsymbol{I}_{m-n} \,\vdots\, -\boldsymbol{C}_{(m-n)\times n} \,\right]$. The use of Eq. (2) is computationally less demanding especially for high-dimensional hierarchical time series. It needs only one matrix inversion of order $(m-n)\times(m-n)$, whereas Eq. (1) needs two matrix inversions of orders $m \times m$ and $n \times n$. Typically in many applications $m - n < n$.

2.2 A quadratic programming solution

To ensure that all entries in $\tilde{\boldsymbol{y}}_T(h)$ are non-negative, it is sufficient to guarantee that all entries in $\tilde{\boldsymbol{b}}_T(h)$ are non-negative. Even though the solution of $\tilde{\boldsymbol{b}}_T(h)$ is derived based on a minimization of the variances of the reconciled forecast errors across the entire structure, it is also apparent from Eq. (1) that $\tilde{\boldsymbol{b}}_T(h)$ is the generalized least squares solution to the following regression problem:

$$\begin{aligned}\min_{\mathring{\boldsymbol{b}}} \frac{1}{2}[\hat{\boldsymbol{y}}_T(h) - \boldsymbol{S}\mathring{\boldsymbol{b}}]^\top \boldsymbol{\Lambda}_h^{-1}[\hat{\boldsymbol{y}}_T(h) - \boldsymbol{S}\mathring{\boldsymbol{b}}] = \\ \min_{\mathring{\boldsymbol{b}}} \frac{1}{2}\mathring{\boldsymbol{b}}^\top\boldsymbol{S}^\top\boldsymbol{\Lambda}_h^{-1}\boldsymbol{S}\mathring{\boldsymbol{b}} - \mathring{\boldsymbol{b}}^\top\boldsymbol{S}^\top\boldsymbol{\Lambda}_h^{-1}\hat{\boldsymbol{y}}_T(h) + \\ \frac{1}{2}\hat{\boldsymbol{y}}_T(h)^\top\boldsymbol{\Lambda}_h^{-1}\hat{\boldsymbol{y}}_T(h).\end{aligned}$$

This suggests that the non-negativity issue can be handled by solving the following quadratic programming problem:

$$\min_{\mathring{\boldsymbol{b}}} q(\mathring{\boldsymbol{b}}) \coloneqq \min_{\mathring{\boldsymbol{b}}} \frac{1}{2}\mathring{\boldsymbol{b}}^\top \boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S}\mathring{\boldsymbol{b}} - \mathring{\boldsymbol{b}}^\top \boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \hat{\boldsymbol{y}}_T(h) \quad \text{s.t.} \quad \mathring{\boldsymbol{b}} \geq \boldsymbol{0}, \tag{3}$$

where the inequalities in Eq. (3) have to hold componentwise. The final non-negative reconciled forecasts are then

$$\breve{\boldsymbol{y}}_T(h) = \boldsymbol{S}\breve{\boldsymbol{b}}_T(h),$$

where $\breve{\boldsymbol{b}}_T(h)$ is the solution to the quadratic programming problem in Eq. (3). This estimation problem is also referred to as "non-negative least squares" (NNLS).

There are a few important features of this minimization problem that are worth discussing. Consider the following statements:

> A vector $\mathring{\boldsymbol{b}}$ is said to be feasible if it satisfies all of the constraints in the quadratic programming problem in Eq. (3). The feasible region is the set of all feasible vectors $\mathring{\boldsymbol{b}}$, and the quadratic programming problem is said to be feasible if the feasible region is non-empty.

> If $\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S}$ is a positive definite matrix, i.e., $\boldsymbol{x}^\top (\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S})\boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq \boldsymbol{0}$, then the objective function of the minimization problem in Eq. (3) is a strictly convex function.

It is easy to show that $\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S}$ is a positive definite matrix by using the fact that the matrix $\boldsymbol{S}$ is of full column rank and assuming that $\boldsymbol{\Lambda}_h$ is positive definite. These simple non-negativity constraints will also ensure that the feasible region is non-empty. Therefore, the quadratic programming problem in Eq. (3) has a unique global solution; i.e., there are no local minima apart from the global minimum (Turlach and Wright, 2015).

Unlike Wickramasuriya, Athanasopoulos, and Hyndman (2019), we will not impose a constraint of unbiasedness on the reconciled forecasts obtained as the solution of the minimization problem in Eq. (3).

The well-known quadratic programming problem in Eq. (3) is easy to solve for small scale hierarchical or grouped structures using the `quadprog` package for R, which is designed to handle dense matrices (Turlach and Weingessel, 2019). However, we require matrices to be stored in a sparse format due to the large size of the structures that typically arise in forecast reconciliation.

2.3 First-order optimality conditions

We can derive first-order necessary conditions for $\breve{\boldsymbol{b}}_T(h)$ to minimize the NNLS problem.

Consider the Lagrangian function for the minimization problem in Eq. (3):

$$\mathcal{L}(\mathring{\boldsymbol{b}}, \boldsymbol{\lambda}) = q(\mathring{\boldsymbol{b}}) - \boldsymbol{\lambda}^\top \mathring{\boldsymbol{b}},$$

where $q(\mathring{\boldsymbol{b}}) = \frac{1}{2}\mathring{\boldsymbol{b}}^\top \boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S}\mathring{\boldsymbol{b}} - \mathring{\boldsymbol{b}}^\top \boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \hat{\boldsymbol{y}}_T(h)$ and $\boldsymbol{\lambda}$ is a Lagrange multiplier vector. The Karush-Kuhn-Tucker (KKT) optimality conditions that need to be satisfied by $\breve{\boldsymbol{b}}_T(h)$ are

$$\begin{aligned}
\nabla_{\mathring{\boldsymbol{b}}} \mathcal{L}[\breve{\boldsymbol{b}}_T(h), \boldsymbol{\lambda}^*] &= \boldsymbol{0}, && && \text{(4a)}\\
\breve{b}_{T,i}(h) &= 0, && \forall i \in \mathcal{A}[\breve{\boldsymbol{b}}_T(h)], && \text{(4b)}\\
\breve{b}_{T,i}(h) &> 0, && \forall i \notin \mathcal{A}[\breve{\boldsymbol{b}}_T(h)], && \text{(4c)}\\
\lambda_i^* &\geq 0, && \forall i \in \mathcal{A}[\breve{\boldsymbol{b}}_T(h)], && \text{(4d)}\\
\lambda_i^* &= 0, && \forall i \notin \mathcal{A}[\breve{\boldsymbol{b}}_T(h)], && \text{(4e)}\\
\lambda_i^* \breve{b}_{T,i}(h) &= 0, && \forall i \in \{1, 2, \ldots, n\}, && \text{(4f)}
\end{aligned}$$

where $\breve{b}_{T,i}(h)$ is the ith component of $\breve{\boldsymbol{b}}_T(h)$ and $\mathcal{A}[\breve{\boldsymbol{b}}_T(h)]$ is referred to as the active set, and is defined as

$$\mathcal{A}[\breve{\boldsymbol{b}}_T(h)] = \left\{ i \in (1, 2, \ldots, n) \;\middle|\; \breve{b}_{T,i}(h) = 0 \right\}.$$

The first optimality condition in Eq. (4a) leads to $\boldsymbol{\lambda}^*$ being computed as

$$\boldsymbol{\lambda}^* = \nabla q[\breve{\boldsymbol{b}}_T(h)] = \boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S}\breve{\boldsymbol{b}}_T(h) - \boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \hat{\boldsymbol{y}}_T(h).$$

The conditions given in Eq. (4f) are referred as complementarity conditions. They indicate that at the optimal solution, either the ith constraint is active or $\lambda_i^* = 0$, or both. Specifically, it implies that the Lagrange multipliers that are associated with inactive inequality constraints are zero.

3 Algorithms

Since the pioneering work of Lawson and Hanson (1974), a variety of methods for solving NNLS problems have been proposed. The following sections briefly explain a few of the algorithms that are suitable for solving large-scale problems, and how they would apply to the forecast reconciliation problem discussed here. A detailed review of methods for NNLS is given by Chen and Plemmons (2009).

3.1 Block principal pivoting method

The first widely used active set method for solving NNLS was that proposed by Lawson and Hanson (1974). The basic idea of this method is to transform the inequality constrained least squares problem into a sequence of equality constrained problems.

Points on the boundary of the feasible region are denoted the "active set", while the remaining points (within the feasible region) are the "passive set". A shortcoming of the standard active set method is that the algorithm is initialized with an empty passive set, and only one variable is added from the active set to the passive set at each step. Although QR updating and down-dating techniques are used to speed up the computations, more iterations (in other words, more time) might be required to find the optimal active set when handling large scale NNLS problems.

One possibility to enhance the speed of the active set method involves including more than one variable from the active set. This should be handled carefully, as it could lead to endless loops in the algorithm. Thus, these types of methods, which are referred to as "block principal pivoting methods", include a procedure for selecting a group of variables to exchange, and a backup rule in order to ensure the finite termination of the algorithm (Júdice and Pires, 1994).

The procedure begins with the monotone linear complementarity problem (LCP) induced by the KKT optimal conditions given in Eq. (4), which needs to be satisfied by the optimal solution.

The monotone linear complementarity problem is

$$\mathring{\boldsymbol{g}} = \boldsymbol{S}^\top [\boldsymbol{\Lambda}_h^{-1} \boldsymbol{S} \mathring{\boldsymbol{b}} - \boldsymbol{\Lambda}_h^{-1} \hat{\boldsymbol{y}}_T(h)], \tag{5}$$

$$\mathring{\boldsymbol{g}} \geq \boldsymbol{0}, \tag{6}$$

$$\mathring{\boldsymbol{b}} \geq \boldsymbol{0}, \tag{7}$$

$$\mathring{b}_i \mathring{g}_i = 0, \quad i = 1, 2, \ldots, n, \tag{8}$$

where $\mathring{b}_i$ and $\mathring{g}_i$ are the ith components of the vectors $\mathring{\boldsymbol{b}}$ and $\mathring{\boldsymbol{g}}$ respectively. A point $(\mathring{\boldsymbol{b}}, \mathring{\boldsymbol{g}}) \in \mathbb{R}^{2n}$ is defined as a complementary solution if it satisfies Eq. (5) and (8).

Let the index set $\{1, 2, \ldots, n\}$ be partitioned into two mutually exclusive subsets F and G. Consider the partitions of $\mathring{\boldsymbol{b}}, \mathring{\boldsymbol{g}}$ and $\boldsymbol{S}$ according to the index sets F and G using the following notation:

$$\mathring{\boldsymbol{b}}_F = [\mathring{b}_i]_{i \in F}, \quad \mathring{\boldsymbol{g}}_F = [\mathring{g}_i]_{i \in F}, \quad \boldsymbol{S}_F = [\boldsymbol{S}_i]_{i \in F},$$

$$\mathring{\boldsymbol{b}}_G = [\mathring{b}_i]_{i \in G}, \quad \mathring{\boldsymbol{g}}_G = [\mathring{g}_i]_{i \in G}, \quad \boldsymbol{S}_G = [\boldsymbol{S}_i]_{i \in G},$$

where $\boldsymbol{S}_i$ is the ith column of $\boldsymbol{S}$.

The algorithm starts by assigning $\mathring{\boldsymbol{b}}_G = \boldsymbol{0}$ and $\mathring{\boldsymbol{g}}_F = \boldsymbol{0}$. This particular choice will ensure that Eq. (8) is always satisfied for any values of $\mathring{\boldsymbol{b}}_F$ and $\mathring{\boldsymbol{g}}_G$. The computation of the remaining unknown quantities $\mathring{\boldsymbol{b}}_F$ and $\mathring{\boldsymbol{g}}_G$ can be carried out by solving the following unconstrained least squares problem:

$$\bar{\boldsymbol{b}}_F = \min_{\mathring{\boldsymbol{b}}_F} \frac{1}{2} \left\| \boldsymbol{S}_F \mathring{\boldsymbol{b}}_F - \hat{\boldsymbol{y}}_T(h) \right\|^2_{\boldsymbol{\Lambda}_h^{-1}}, \tag{9}$$

and then setting

$$\bar{\boldsymbol{g}}_G = \boldsymbol{S}_G^\top [\boldsymbol{\Lambda}_h^{-1} \boldsymbol{S}_F \bar{\boldsymbol{b}}_F - \boldsymbol{\Lambda}_h^{-1} \hat{\boldsymbol{y}}_T(h)], \tag{10}$$

where $\|.\|_{\boldsymbol{\Lambda}_h^{-1}} = [\boldsymbol{S}_F \mathring{\boldsymbol{b}}_F - \hat{\boldsymbol{y}}_T(h)]^\top \boldsymbol{\Lambda}_h^{-1} [\boldsymbol{S}_F \mathring{\boldsymbol{b}}_F - \hat{\boldsymbol{y}}_T(h)]$. The solution pair $(\bar{\boldsymbol{b}}_F, \bar{\boldsymbol{g}}_G)$ is referred to as a complementary basic solution. If a complementary basic solution satisfies $\bar{\boldsymbol{b}}_F \geq \boldsymbol{0}$ and $\bar{\boldsymbol{g}}_G \geq \boldsymbol{0}$, then we have reached at the optimal solution of the NNLS problem. Otherwise, the complementary basic solution is infeasible; i.e., there exists at least one $i \in F$ with $\bar{b}_i < 0$ or one $i \in G$ with $\bar{g}_i < 0$.

In the presence of an infeasible solution, we need to update sets F and G by exchanging the variables for which Eqs. (6) or (7) do not hold. Define the following two sets, which correspond to the infeasibilities in sets F and G:

$$I_1 = \{i \in F : \bar{b}_i < \varepsilon\} \qquad \text{and} \qquad I_2 = \{i \in G : \bar{g}_i < \varepsilon\}, \tag{11}$$

where $\varepsilon = 10^{-12}$. For a given $\bar{I}_1 \subseteq I_1$ and $\bar{I}_2 \subseteq I_2$, F and G can be updated according to the following rules:

$$F = (F - \bar{I}_1) \cup \bar{I}_2 \qquad \text{and} \qquad G = (G - \bar{I}_2) \cup \bar{I}_1. \tag{12}$$

The first block principal pivoting algorithm for solving a strictly monotonic LCP is due to the work of Kostreva (1978). Unfortunately, the use of blocks of variables for exchange can lead to a cycle, meaning that it is not guaranteed to provide the optimal solution. Although this occurs rarely, it is problematic (Júdice and Pires, 1989). In a later paper, Júdice and Pires (1994) proposed an extension of this algorithm to include finite termination, by incorporating Murty's single principal pivoting algorithm. Even though the algorithm proposed by Murty (1974) has finite termination, convergence can be slow for applications with large numbers of variables, as it changes only one variable per iteration. However, this algorithm is still beneficial for obtaining a complementary basic solution with a smaller number of infeasibilities than before. The steps of the hybrid algorithm are given in Algorithm 3.1. card(X) denotes the cardinality of set X.

In particular, if $\boldsymbol{\Lambda}_h$ is a diagonal matrix with positive elements and $\hat{\boldsymbol{y}}_T(h) > \boldsymbol{0}$, Algorithm 3.1 can start by defining the initial conditions as $F = \{1, 2, \ldots, n\}$, $G = \emptyset$, $\mathring{\boldsymbol{b}} = \tilde{\boldsymbol{b}}$, $\mathring{\boldsymbol{g}} = \boldsymbol{0}$, where $\tilde{\boldsymbol{b}}$ is the original unconstrained

Algorithm 3.1 Block principal pivoting algorithm

INITIALIZATION: Let $F = \emptyset$, $G = \{1, 2, \ldots, n\}$, $\mathring{\boldsymbol{b}} = \mathbf{0}$, $\mathring{\boldsymbol{g}} = -\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \hat{\boldsymbol{y}}_T(h)$, $p = \bar{p} \leq 10$, and ninf $= n + 1$, and let α be a permutation of the set $\{1, 2, \ldots, n\}$.

1: IF ($\mathring{\boldsymbol{b}}_F \geq \mathbf{0}$ & $\mathring{\boldsymbol{g}}_G \geq \mathbf{0}$) THEN
2: Terminate the algorithm and $\breve{\boldsymbol{b}} = (\mathring{\boldsymbol{b}}_F, \mathbf{0})$ is the unique global solution.
3: ELSE
4: Define I_1 and I_2 as given in Eq. (11).
5: IF (card$(I_1 \cup I_2) <$ ninf) THEN
6: Set ninf = card$(I_1 \cup I_2)$, $p = \bar{p}$, $\bar{I}_1 = I_1$ and $\bar{I}_2 = I_2$.
7: ELSEIF (card$(I_1 \cup I_2) \geq$ ninf and $p \geq 1$) THEN
8: Set $p = p - 1$, $\bar{I}_1 = I_1$ and $\bar{I}_2 = I_2$.
9: ELSEIF (card$(I_1 \cup I_2) \geq$ ninf and $p = 0$) THEN
10: Set $\bar{I}_1 = \{r\}$ and $\bar{I}_2 = \emptyset$, if $r \in I_1$,
11: $\bar{I}_1 = \emptyset$ and $\bar{I}_2 = \{r\}$, if $r \in I_2$,
12: where r is the last element of the set $I_1 \cup I_2$ as for the order defined by α.
13: ENDIF
14: Update F and G as given by Eq. (12).
15: Compute $\bar{\boldsymbol{b}}_F$ and $\bar{\boldsymbol{g}}_G$ using Eqs. (9) and (10) respectively, and assign $\mathring{\boldsymbol{b}}_F = \bar{\boldsymbol{b}}_F$ and $\mathring{\boldsymbol{g}}_G = \bar{\boldsymbol{g}}_G$.
16: Return to line 1.
17: ENDIF

least squares solution with positive diagonal elements for $\boldsymbol{\Lambda}_h$.

Rather than selecting a subset of variables from I_1 and I_2, we speed up the computations by using the full sets of variables as $\bar{I}_1$ and $\bar{I}_2$ respectively. This is generally referred to as the "full exchange rule". The variable p in Algorithm 3.1 acts as a buffer for determining the number of full exchange rules that may be tried. The choice of p plays an important rule. It should be fairly small in order to prevent unnecessary computations in which the full exchange rule is not effective. However, it should not be too small, otherwise Murty's method may be activated several times, thus reducing the efficiency of the algorithm. In general, $p = 3$ is a good choice (Júdice and Pires, 1994).

3.2 Gradient projection + conjugate gradient approach

Each iteration of this algorithm is designed to follow two main steps. In the first step, the current feasible solution $\mathring{\boldsymbol{b}}$ is updated by searching along the steepest direction; in other words, the direction $-\mathring{\boldsymbol{g}}$ from $\mathring{\boldsymbol{b}}$. If the lower bound of the inequality constraints (i.e., $\mathbf{0}$) is encountered before a minimizer is found along the line, the search direction is "bent" to ensure that it remains feasible. The search is continued along the resulting piecewise-linear path in order to locate the first local minimizer of the objective function, q. This point is referred to as the Cauchy point. Based on the Cauchy point, it is possible to define a set of constraints that are active at this point. Hence, in the second step, a subproblem is solved by fixing the constraints of the active set to zero. The key steps of this method are given in Algorithm 3.2. Refer Nocedal and Wright (2006) for a detailed implementation of each step involved.

Algorithm 3.2 Gradient projection based on the Cauchy point

INITIALIZATION: Choose a feasible initial solution $\mathring{\boldsymbol{b}}^0$.

1: FOR k in 0, 1, 2, ... DO
2: IF $\mathring{\boldsymbol{b}}^k$ satisfies the KKT conditions THEN
3: Terminate the algorithm. $\breve{\boldsymbol{b}} = \mathring{\boldsymbol{b}}^k$ is the unique global solution.
4: ELSE
5: Find the Cauchy point $\boldsymbol{b}^c$ using $\mathring{\boldsymbol{b}}^k$.
6: Use projected conjugate gradient algorithm with a diagonal preconditioner to find an approximate feasible solution $\mathring{\boldsymbol{b}}^+$ that satisfies $q(\mathring{\boldsymbol{b}}^+) \leq q(\boldsymbol{b}^c)$.
7: $\mathring{\boldsymbol{b}}^{k+1} = \mathring{\boldsymbol{b}}^+$.
8: ENDIF
9: ENDFOR

3.3 Scaled gradient projection

The diagonally scaled gradient projection algorithm was proposed by Bonettini, Zanella, and Zanni (2009). The algorithm propagates by determining a descent direction at each iteration, based on the current feasible solution, step length and scaling matrix. The solution vector is then adjusted along this direction using a non-monotone line search that does not guarantee a decrease

Algorithm 3.3 Diagonally scaled gradient projection algorithm

INITIALIZATION: Choose a feasible initial solution $\mathring{\boldsymbol{b}}^0$. Set the parameters $\eta, \theta \in (0,1)$, $0 < \alpha_{\min} < \alpha_{\max}$, and a positive integer M. Use a diagonal scaling matrix $\boldsymbol{D} = \text{diag}(d_1, d_2, \ldots, d_n)$ with elements $d_i = 1/(\boldsymbol{S}^\top \boldsymbol{\Lambda}_h^{-1} \boldsymbol{S})_{ii}$ for $i = 1, 2, \ldots, n$.

1: FOR k in 0, 1, 2, ... DO
2: Choose the step-length parameter $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ using the alternation strategy proposed by Bonettini, Zanella, and Zanni (2009).
3: Projection: $\boldsymbol{z}^k = [\mathring{\boldsymbol{b}}^k - \alpha_k \boldsymbol{D} \mathring{\boldsymbol{g}}(\boldsymbol{b}^k)]_+$, where $(x)_+ = \max(0, x)$.
4: IF $\boldsymbol{z}^k = \mathring{\boldsymbol{b}}^k$ THEN
5: Terminate the algorithm. $\breve{\boldsymbol{b}} = \boldsymbol{b}^k$ is the unique global minimum.
6: ELSE
7: Descent direction: $\boldsymbol{d}^k = \boldsymbol{z}^k - \mathring{\boldsymbol{b}}^k$.
8: Set $\lambda_k = 1$ and $q_{\max} = \max_{j \in 0,1,\ldots,\min(k, M-1)} q(\mathring{\boldsymbol{b}}^{k-j})$
9: Backtracking loop:
IF $q(\mathring{\boldsymbol{b}}^k + \lambda_k \boldsymbol{d}^k) \leq q_{\max} + \eta \lambda_k [\mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^k)]^\top \boldsymbol{d}^k$ THEN
Go to line 10.
ELSE
Set $\lambda_k = \theta \lambda_k$ and go to line 9.
ENDIF
10: Set $\mathring{\boldsymbol{b}}^{k+1} = \mathring{\boldsymbol{b}}^k + \lambda_k \boldsymbol{d}^k$.
11: ENDIF
12: ENDFOR

in the objective function value at each iteration. This is in order to increase the likelihood of locating a global optimum in practice (Birgin, Martínez, and Raydan, 2003). The main steps are given in Algorithm 3.3.

In selecting the step-length parameter for the scaled gradient projection algorithm, Bonettini, Zanella, and Zanni (2009) extended the original ideas of Barzilai and Borwein (1988) that are commonly used for improving the convergence rate of standard gradient methods. The generalized Barzilai & Borwein (BB) rules are

$$\alpha_k^{BB1} = \frac{(\boldsymbol{u}^{k-1})^\top \boldsymbol{D}^{-1} \boldsymbol{D}^{-1} \boldsymbol{u}^{k-1}}{(\boldsymbol{u}^{k-1})^\top \boldsymbol{D}^{-1} \boldsymbol{v}^{k-1}},$$

$$\alpha_k^{BB2} = \frac{(\boldsymbol{u}^{k-1})^\top \boldsymbol{D} \boldsymbol{v}^{k-1}}{(\boldsymbol{v}^{k-1})^\top \boldsymbol{D} \boldsymbol{D} \boldsymbol{v}^{k-1}},$$

where $\boldsymbol{u}^{k-1} = \mathring{\boldsymbol{b}}^k - \mathring{\boldsymbol{b}}^{k-1}$ and $\boldsymbol{v}^{k-1} = \mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^k) - \mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^{k-1})$. Specifically, these equations reduce to the standard BB rules when $\boldsymbol{D} = \boldsymbol{I}_n$.

When $[\mathring{\boldsymbol{b}}^k - \alpha_k \boldsymbol{D} \mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^k)]_+$ is used to generate a descent direction from $\mathring{\boldsymbol{b}}^k$, the diagonal scaling matrix with $\boldsymbol{\Lambda}_h \propto \boldsymbol{I}_n$ and the generalized BB rules indicate that the whole process reduces to the use of a standard gradient projection method with standard BB rules. Thus, the scaled gradient projection algorithm might not be computationally advantageous for obtaining non-negative reconciled forecasts from the OLS approach proposed by Hyndman et al. (2011).

3.3.1 Selection of tuning parameters

We now discuss the roles of the tuning parameters and their recommended values.

– α_0: We consider a method similar to that of Figueiredo, Nowak, and Wright (2007), implemented in standard gradient projection methods, within the context of the scaled gradient projection methods. It considers the initial value α_0 as the exact minimizer of the objective function along the direction of $\mathring{\boldsymbol{b}}^0 - \alpha \boldsymbol{D} \mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^0)$, if no constraints are to be satisfied. This involves defining

$$\boldsymbol{p}^0 = (p_i^0) = \begin{cases} [\mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^0)]_i, \text{ if } \mathring{b}_i^0 > 0 \text{ or } [\mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^0)]_i < 0, \\ 0, \qquad \text{otherwise.} \end{cases}$$

The initial guess is estimated as

$$\alpha_0 = \min_{\alpha} q[\mathring{\boldsymbol{b}}^0 - \alpha \boldsymbol{D}\mathring{\boldsymbol{g}}(\mathring{\boldsymbol{b}}^0)],$$

and can be computed analytically using

$$\alpha_0 = \frac{(\boldsymbol{p}^0)^\top \boldsymbol{D}\boldsymbol{p}^0}{(\boldsymbol{S}\boldsymbol{D}\boldsymbol{p}^0)^\top \boldsymbol{\Lambda}_h^{-1}(\boldsymbol{S}\boldsymbol{D}\boldsymbol{p}^0)}.$$

- η and θ control the amount by which the objective function should be decreased and the number of backtracking reductions to be performed, respectively. The values of $\eta = 10^{-4}$ and $\theta = 0.4$ have been used in order to get a sufficiently large step size with fewer reductions (Bonettini, Zanella, and Zanni, 2009).
- $\alpha_{\min}$ and $\alpha_{\max}$ are the lower and upper bounds of the step-length parameter α_k, to avoid using unnecessary extreme values. Even though a large range is defined for the BB-type rules in practice, Bertero et al. (2013) found the interval $(10^{-5}, 10^5)$ to be suitable for the generalized BB rules.
- τ_1 is the initial switching condition that activates the step-length alternation strategy, and a value of 0.5 is suitable in many applications (Bertero et al., 2013; Bonettini, Zanella, and Zanni, 2009).
- We set $M_\alpha = 3$, as was used by Bonettini, Zanella, and Zanni (2009) and Bertero et al. (2013).
- M determines the monotone ($M = 1$) or non-monotone ($M > 1$) line search to be performed in the backtracking loop. This value should not be too large, as the decrease in the objective function is difficult to control, and is set to 10 here, as was done by Bonettini, Zanella, and Zanni (2009).

4 Monte Carlo experiments

Monte Carlo experiments can help demonstrate the practical usefulness of the aforementioned algorithms for obtaining a set of non-negative reconciled forecasts. For the sake of simplicity, the WLS based on structural weights (WLS_s) approach is considered (Wickramasuriya, Athanasopoulos, and Hyndman, 2019). The computational efficiency of these algorithms is evaluated over a series of hierarchies, ranging in size from small to large. We study the behaviours of two possible choices of the initial solution: (i) base forecasts at the bottom level; and (ii) the unconstrained WLS_s forecasts. The latter choice can sometimes be a better alternative than the former, as it is aggregate consistent, and computationally less demanding. For the projection-based approaches, the unconstrained WLS_s forecasts are projected on to the non-negative orthant, as these algorithms need a feasible initial solution. However, the block principal pivoting algorithm can use the unconstrained solution in its original condition. The acronyms used to distinguish the algorithms and their variations of interest are listed in Table 1.

Table 1: Acronyms for NNLS algorithms.

Algorithm	Notation
Scaled gradient projection	SGP
Gradient projection + conjugate gradient	PCG
Block principal pivoting	BPV

All experiments are performed in R on a Linux machine equipped with a 3.20GHz Intel Quad-core processor and 8GB memory. A parallel computing environment with two workers is established by using the `doParallel` (Microsoft Corporation and Weston, 2019a) and `foreach` (Microsoft Corporation and Weston, 2019b) packages in R to parallelize the procedure of computing non-negative reconciled forecasts for different forecast horizons.

Initially, we consider a hierarchy with $K = 1$ level, having three series at the bottom level. The base forecast for the most aggregated series in the hierarchy is generated from a uniform distribution on the interval $(1.5e^K, 2e^K)$, where K is the number of levels in the hierarchy. This is then disaggregated to the bottom level based on a set of proportions that sum to one. Specifically, a set of values is chosen from a gamma distribution, with the shape and scale parameters set to two, and the values are normalized to ensure that they sum to one. Noise is then added to the series at the aggregated levels to make them aggregate-inconsistent. If any of the base forecasts become negative after the noise is added, they are set to zero in order to ensure that all base forecasts in the hierarchy are strictly non-negative. The whole procedure is repeated until there is at least one negative reconciled forecast at the bottom level, and six of these sets with negative reconciled forecasts are used to denote a forecast horizon of length 6. The number of levels in the hierarchy is increased gradually to construct much larger hierarchies, by adding a mixture of three and four nodes to the bottom-level nodes in the preceding hierarchy. Table 2 presents the structure of each hierarchy constructed and the number of negative reconciled forecasts at the bottom level for each forecast horizon. Each hierarchy contains approximately 5–10% of negative reconciled forecasts at the bottom level. These are then revised using the algorithms discussed in Section 3, in order to obtain a set of non-negative reconciled forecasts.

Table 2: The structure of each hierarchy generated and the numbers of negative reconciled forecasts that result from the WLS_s approach.

			Forecast horizon (h)					
K	m	n	1	2	3	4	5	6
1	4	3	1	1	1	1	1	1
2	14	10	1	1	1	2	1	1
3	49	35	1	1	1	1	2	1
4	171	122	1	1	3	5	2	4
5	598	427	13	10	10	9	6	7
6	2092	1494	62	38	60	56	44	43
7	7321	5229	211	229	255	270	222	203
8	25622	18301	1128	1166	1026	1268	1285	1186
9	89675	64053	4738	5619	6244	4812	5779	5637
10	249808	160133	17744	17790	16023	17261	19462	17258
11	650141	400333	36271	47845	44040	40790	47879	38753
12	1650974	1000833	105462	84998	76817	102130	95530	80090

Note: For $K > 9$, either two or three nodes are added to each of the bottom-level nodes in the preceding hierarchy.

Tables 3 and 4 present the numbers of iterations and the average computational times in seconds (s) that are required to reach the KKT optimality conditions when the base and projected (unconstrained) WLS_s forecasts, respectively, are used as the initial solution. Cases where an algorithm reaches the maximum number of iterations (10^4) are marked with an asterisk, and the average computational time corresponding to that number of iterations is given. The first row in each hierarchy of Table 3 gives the computational times required to produce the unconstrained WLS_s reconciled forecasts, which is always the best time, because the weight matrix has an analytical expression and computes only ones for all forecast horizons. The bold entries identify the non-negative algorithms with the best computational performances.

The main conclusion that can be drawn from these sets of results is that the BPV algorithm always has the best computational performance. This is due in part to the alternative analytical representation proposed for MinT (Wickramasuriya, Athanasopoulos, and Hyndman, 2019). In addition, it should be noted that the computational performance of BPV algorithm depends strongly on how often the full exchange rule fails and Murty's method or the back-up rule has to be activated. The back-up rule was inactive for all experiments carried out in this section; this is also observed in the tests performed by Kim and Park (2011). However, there is no theoretical justification for this, nor do we have conditions under which we know that the back-up rule is always inactive. Hence, the performance of the algorithm will be affected slightly if it is activated in a certain application.

The second best timing is achieved by the PCG algorithm. Unfortunately, it is inefficient for larger hierarchies, as locating the Cauchy point can be time consuming. Of the two initial solutions considered, the (projected) unconstrained WLS_s is a good choice, as expected.

We also evaluated the performance of these algorithms for obtaining non-negative reconciled forecasts using the OLS approach. The BPV and PCG algorithms showed similar performances as those observed with WLS_s. The SGP algorithm performed the worst; this is not surprising, as it reduces to using the standard gradient projection algorithm for the OLS approach.

To further examine the superiority of these competing algorithms on real applications, an extensive case study is provided in Section 5.

5 Non-negative reconciled forecasts for Australian domestic tourism flows

This section evaluates the impact of imposing the strict non-negativity constraints on the forecast reconciliation approaches, using the Australian domestic tourism flows as a case study. We consider a grouped structure comprising 555 series. A detailed description of the structure and a comprehensive analysis of the forecast performances of different reconciliation approaches are given in Wickramasuriya, Athanasopoulos, and Hynd-

Table 3: Computational efficiency of the non-negative forecast reconciliation from the WLS_s approach using base forecasts as the initial solution.

				Forecast horizon (h)						
K	m	n		1	2	3	4	5	6	Time (s)
1	4	3	WLS_s							0.01
			SGP	3	3	3	3	3	3	**0.04**
			PCG	1	1	1	1	1	1	**0.04**
2	14	10	WLS_s							0.01
			SGP	342	265	23	481	22	497	1.52
			PCG	1	2	1	1	1	2	**0.10**
3	49	35	WLS_s							0.01
			SGP	276	461	391	392	737	423	1.86
			PCG	2	2	2	1	1	2	**0.17**
4	171	122	WLS_s							0.01
			SGP	456	496	389	480	364	498	2.13
			PCG	2	2	2	2	2	2	**0.29**
5	598	427	WLS_s							0.01
			SGP	375	357	379	352	359	270	1.94
			PCG	2	3	2	3	2	3	**0.33**
6	2092	1494	WLS_s							0.02
			SGP	1239	354	419	7803	354	382	29.11
			PCG	5	6	6	6	4	6	**0.51**
7	7231	5229	WLS_s							0.06
			SGP	1020	851	10^{4*}	10^{4*}	614	369	103.16
			PCG	9	8	8	9	7	7	**2.09**
8	25622	18301	WLS_s							0.19
			PCG	13	14	13	13	13	12	**19.08**
9	89675	64053	WLS_s							0.48
			PCG	16	16	16	17	17	21	**244.68**
10	249808	160133	WLS_s							1.43
			PCG	19	20	19	17	20	20	**1660.12**

Notes: WLS_s defines the unconstrained WLS_s approach.
The computational time is averaged over 50 replications for $K = 1$ to $K = 9$, but only 10 for $K = 10$, as the computational time is considerable.
Only PCG is performed up to $K = 10$, due to the high computational time.

man (2019). However, the base and reconciled forecasts in these applications were allowed to take any values on the real line; in other words, they were not explicitly restricted to be non-negative, even though the original data are.

The empirical study performed in Wickramasuriya, Athanasopoulos, and Hyndman (2019) is repeated by log-transforming the original data. For the series that include zeros, a small constant — specifically, half of the minimum positive value — is added and aggregates computed before transforming the data. Then ARIMA and ETS models are fitted by minimizing the AICc to obtain the base forecasts using the default settings as implemented in the `forecast` package for R (Hyndman, 2019) and described in Hyndman and Khandakar (2008). If the forecast densities on the transformed data are symmetric, then the back-transformed base forecasts give the median of the forecast density. The mean of the back-transformed forecasts is given by $\exp\left(\mu + \sigma^2/2\right)$, where μ and σ^2 are the mean and variance on the transformed scale, respectively (Johnson, Kotz, and Balakrishnan, 1994, p. 212).

These back-transformed means (base forecasts) are then reconciled using OLS, WLS based on variance scaling (WLS_v), and MinT. MinT uses the covariance estimator that shrinks the sample correlation matrix towards an identity matrix; refer to Wickramasuriya, Athanasopoulos, and Hyndman (2019) for further details about covariance estimators. The expanding window forecast evaluation procedure results in a total of 96 1-step-ahead reconciled forecasts, 95 2-step-ahead reconciled forecasts, and so on, down to 85 12-step-ahead reconciled forecasts. These include iterations in which all reconciled forecasts are positive. Table 5 presents the summary statistics of the negative reconciled forecasts obtained from the ARIMA and ETS base fore-

Table 4: Computational efficiency of the non-negative forecast reconciliation from the WLS_s approach using (projected) unconstrained WLS_s forecasts as the initial solution.

				Forecast horizon (h)						
K	m	n		1	2	3	4	5	6	Time (s)
1	4	3	SGP	1	1	1	1	1	1	**0.03**
			PCG	1	1	1	1	1	1	**0.03**
			BPV	1	1	1	1	1	1	0.06
2	14	10	SGP	287	17	22	512	357	35	0.88
			PCG	1	1	1	1	1	2	0.09
			BPV	1	1	1	1	1	2	**0.08**
3	49	35	SGP	290	347	297	374	621	388	1.80
			PCG	1	1	1	1	1	1	0.13
			BPV	1	1	1	1	1	1	**0.07**
4	171	122	SGP	515	312	423	431	368	12	2.02
			PCG	1	1	1	1	1	1	0.17
			BPV	1	1	1	1	1	1	**0.09**
5	598	427	SGP	455	380	363	334	419	368	2.27
			PCG	1	1	1	1	1	1	0.22
			BPV	1	1	1	1	2	1	**0.12**
6	2092	1494	SGP	1214	282	424	5947	390	340	22.32
			PCG	3	2	2	2	2	1	0.36
			BPV	2	2	1	2	2	1	**0.17**
7	7321	5229	SGP	879	663	10^{4*}	10^{4*}	766	374	101.72
			PCG	3	4	6	4	3	3	0.68
			BPV	2	2	2	2	2	2	**0.57**
8	25622	18301	PCG	6	8	8	7	6	5	3.18
			BPV	2	2	2	2	3	2	**1.76**
9	89675	64053	PCG	12	11	14	11	10	14	35.08
			BPV	3	3	3	3	3	3	**6.45**
10	249808	160133	PCG	16	16	18	17	17	16	250.22
			BPV	3	3	3	3	3	3	**21.00**
11	650141	400333	PCG	18	19	21	22	18	21	1597.10
			BPV	3	3	3	3	3	3	**56.84**
12	1650974	1000833	BPV	3	3	3	3	3	3	**3247.09**

Notes: The computational time is averaged over 50 replications for $K = 1$ to $K = 9$, but only 10 for $K > 9$, as the computational time is considerable.
Only PCG is performed up to $K = 11$, due to the high computational time.

casts. Each cell gives (i) the number of iterations reported with negative reconciled forecasts; and the minimum and maximum of (ii) the number of negative reconciled forecasts; (iii) the largest negative value (in thousands); and (iv) the smallest negative value (in thousands) resulting from each of the forecast reconciliation approaches. For example, the first cell in the table illustrates that all 96 iterations of the 1-step-ahead OLS reconciled forecasts include negative forecasts, and the number of negatives can be as small as 5 or as large as 80. The largest negative value varies between -71.8 and -2.33, and the smallest negative value varies between -1.52 and -0.002. It is clear that both the number of negative reconciled forecasts and their magnitudes are considerably large with the OLS approach. The presence of such negative values can degrade the quality of the forecasts in the entire structure.

These forecasts are then revised using the algorithms discussed in Section 3. For the scaled gradient projection and projected conjugate gradient algorithms, these forecasts are projected into the non-negative orthant and used as the initial solution. When the MinT approach is coupled with the block principal pivoting algorithm, a vector of zeros is used as the initial solution. Table 6 summarizes the total computational time (in seconds) of each forecast reconciliation approach for obtaining a set of non-negative reconciled forecasts. Cases where an algorithm reaches the maximum number of iterations (10^4) are marked with an asterisk, and the computational time that corresponds to that number of iterations is given. For example, the first cell in the

Table 5: Summary statistics of the negative reconciled forecasts.

	ARIMA						ETS					
	OLS		WLS_v		MinT		OLS		WLS_v		MinT	
$h=1$		96		39		61		93		45		43
	5	80	1	4	1	8	1	86	1	3	1	3
	−71.8	−2.33	−17.5	−0.07	−22.8	−0.05	−41.0	−0.03	−13.2	−0.02	−14.9	−0.01
	−1.52	−2e-3	−13.2	−0.06	−10.9	−4e-3	−6.16	−4e-3	−12.7	−0.02	−9.78	−0.01
2		95		37		64		93		43		45
	5	91	1	3	1	9	1	63	1	3	1	3
	−72.3	−2.6	−12.9	−0.09	−23.1	−0.08	−38.4	−0.19	−13.2	−0.11	−16.9	−0.10
	−3.98	−1e-3	−10.7	−0.09	−18.3	−0.01	−7.00	−1e-3	−12.0	−0.07	−8.80	−0.09
3		94		37		62		91		46		46
	5	90	1	2	1	8	1	66	1	3	1	3
	−67.3	−1.31	−16.3	−0.04	−14.5	−0.02	−34.0	−0.12	−11.7	−0.08	−13.4	−0.02
	−2.36	−3e-4	−16.3	−0.04	−14.5	−0.02	−4.58	−2e-3	−11.7	−0.08	−8.80	−3e-3
6		91		40		56		90		40		43
	4	87	1	3	1	9	1	75	1	3	1	3
	−87.4	−0.62	−12.3	−0.11	−16.5	−0.03	−31.8	−0.15	−15.2	−3e-4	−13.4	−0.17
	−1.16	−4e-4	−9.91	−0.11	−10.49	−0.03	−4.40	−1e-3	−15.2	−3e-4	−8.12	−1e-3
12		85		44		59		84		37		39
	5	83	1	2	1	8	1	117	1	3	1	4
	−79.5	−2.30	−19.2	−0.18	−26.4	−0.07	−170	−0.15	−15.2	−0.07	−17.4	−0.10
	−3.66	−5e-4	−19.2	−0.18	−15.6	−5e-3	−3.44	−3e-3	−14.1	−0.04	−10.5	−0.05

Notes: Each cell gives (i) the number of iterations reported with negative reconciled forecasts; and the minimum and maximum of (ii) the number of negative reconciled forecasts; (iii) the largest negative value (in thousands); and (iv) the smallest negative value (in thousands).

Table 6: Computational times (in seconds) of obtaining the non-negative reconciled forecasts.

		$h=1$	*2*	*3*	*6*	*12*	$h=1$	*2*	*3*	*6*	*12*
		ARIMA					ETS				
OLS	SGP	971.1*	953.7*	943.8*	912.4*	858.2*	947.5*	918.4*	897.5*	885.7*	828.6*
	PCG	27.6	27.5	27.6	27.0	24.2	23.0	22.8	21.8	21.4	21.1
	BPV	**3.0**	**2.8**	**2.9**	**2.8**	**2.6**	**2.3**	**2.3**	**2.2**	**2.1**	**2.0**
WLS_v	SGP	**2.0**	**1.9**	**1.9**	**1.9**	**1.8**	**2.1**	**2.0**	**2.0**	**1.9**	**1.8**
	PCG	5.2	5.0	4.8	5.0	5.5	5.7	5.5	5.6	5.0	4.8
	BPV	2.2	2.1	2.1	2.0	2.0	2.2	2.1	2.2	2.0	1.9
MinT	SGP	54.4	55.2	55.2	50.5	48.7	46.8	47.1	46.8	44.7	41.4
	PCG	50.1	49.9	49.1	46.5	44.7	45.1	44.9	44.6	42.9	39.8
	BPV	**38.4**	**37.8**	**37.5**	**36.2**	**33.7**	**38.2**	**37.7**	**37.3**	**36.0**	**33.6**

Notes: The bold entries identify the non-negative algorithms with the best computational performances.

table illustrates that the SGP algorithm takes 971.1 seconds to revise all 96 iterations with negative reconciled forecasts. However, it has failed to reach the optimal solution.

As was observed in the simulation exercise, the BPV algorithm showed the best computational performance for the OLS approach, while PCG was the second best. The worst performance of the scaled gradient projection algorithm fits with the previous findings. The WLS_v approach based on the SGP algorithm showed the best performances, challenging the BPV algorithm.

Table 7: Impact of the non-negativity constraints on forecast performance.

	$h=1$	*2*	*3*	*6*	*12*	$h=1$	*2*	*3*	*6*	*12*
	ARIMA									
	Australia					Australia by purpose of travel				
OLS	0.0081	**−0.0138**	**−0.0230**	**−0.0094**	**−0.0269**	**−0.0757**	**−0.0503**	0.0125	**−0.0065**	**−0.0157**
WLS_v	0.0091	0.0056	0.0100	0.0006	0.0018	0.0009	0.0027	0.0027	0.0031	0.0008
MinT	**−0.0090**	0.0267	0.0290	0.0427	**−0.0222**	0.0295	0.0143	0.0514	0.0597	0.0279
	States					States by purpose of travel				
OLS	**−0.0560**	**−0.0744**	**−0.0543**	**−0.0159**	**−0.0904**	**−0.5054**	**−0.8173**	**−0.7670**	**−0.8597**	**−0.7745**
WLS_v	0.0045	0.0016	0.0004	**−0.0077**	0.0003	**−0.0009**	**−0.0004**	**−0.0013**	**−0.0023**	**−0.0037**
MinT	0.0366	0.0007	0.0123	0.0140	0.0074	0.0160	0.0120	0.0135	0.0255	**−0.0164**
	Zones					Zones by purpose of travel				
OLS	**−0.1115**	**−0.1038**	**−0.1090**	**−0.1141**	**−0.1489**	**−0.4320**	**−0.6144**	**−0.5723**	**−0.7088**	**−0.6646**
WLS_v	0.0015	**−0.0009**	0.0011	0.0019	**−0.0018**	**−0.0004**	0.0005	0.0001	0.0001	**−0.0071**
MinT	0.0028	**−0.0037**	0.0031	0.0019	**−0.0161**	0.0004	**−0.0044**	0.0032	0.0072	**−0.0234**
	Regions					Regions by purpose of travel				
OLS	**−0.1474**	**−0.1246**	**−0.1164**	**−0.1263**	**−0.1480**	**−0.5573**	**−0.6581**	**−0.5831**	**−0.7258**	**−0.7480**
WLS_v	0.0014	0.0017	0.0008	0.0048	0.0114	**−0.0052**	**−0.0049**	**−0.0044**	**−0.0057**	**−0.0150**
MinT	**−0.0002**	**−0.0056**	0.0002	**−0.0003**	**−0.0012**	**−0.0134**	**−0.0189**	**−0.0104**	**−0.0097**	**−0.0265**
	ETS									
	Australia					Australia by purpose of travel				
OLS	**−0.0236**	**−0.0021**	**−0.0028**	**−0.0116**	**−0.0468**	**−0.0066**	**−0.0088**	**−0.0060**	**−0.0103**	**−0.0222**
WLS_v	0.0052	0.0040	0.0044	0.0059	**−0.0002**	**−0.0018**	**−0.0026**	**−0.0023**	**−0.0043**	**−0.0026**
MinT	0.0089	0.0193	0.0207	0.0203	0.0048	**−0.0012**	**−0.0118**	0.0020	**−0.0105**	**−0.0127**
	States					States by purpose of travel				
OLS	**−0.0333**	**−0.0314**	**−0.0207**	**−0.0510**	**−0.2106**	**−0.0636**	**−0.0423**	**−0.0455**	**−0.0593**	**−0.1031**
WLS_v	**−0.0017**	**−0.0016**	0.0009	0.0033	**−0.0009**	0.0026	0.0022	0.0050	0.0040	0.0011
MinT	0.0129	0.0090	0.0144	0.0165	0.0031	0.0101	0.0060	0.0095	0.0056	**−0.0001**
	Zones					Zones by purpose of travel				
OLS	**−0.0274**	**−0.0213**	**−0.0226**	**−0.0350**	**−0.1622**	**−0.0752**	**−0.0632**	**−0.0703**	**−0.0543**	**−0.0962**
WLS_v	**−0.0084**	**−0.0095**	**−0.0120**	**−0.0133**	**−0.0099**	0.0038	0.0033	0.0028	0.0033	0.0034
MinT	**−0.0008**	**−0.0036**	**−0.0057**	**−0.0071**	**−0.0093**	0.0027	**−0.0018**	0.0004	0.0012	0.0014
	Regions					Regions by purpose of travel				
OLS	**−0.0262**	**−0.0239**	**−0.0297**	**−0.0288**	**−0.1026**	**−0.1249**	**−0.0940**	**−0.0897**	**−0.0912**	**−0.1295**
WLS_v	0.0081	0.0060	0.0040	0.0076	0.0085	**−0.0148**	**−0.0131**	**−0.0112**	**−0.0119**	**−0.0140**
MinT	0.0121	0.0102	0.0083	0.0110	0.0135	**−0.0149**	**−0.0148**	**−0.0131**	**−0.0138**	**−0.0152**

Notes: Each entry shows the percentage difference in average RMSEs between reconciled forecasts with non-negatives and negatives. A negative (positive) entry shows a percentage decrease (increase) in average RMSEs relative to the negative reconciled forecasts. Bold entries identify improvements due to the non-negativity constraints.

Even though MinT resulted in fewer negative reconciled forecasts, the computational time is much larger than the best timings of the WLS_v and even OLS approaches using the BPV and PCG algorithms. This is reasonable because a full variance covariance matrix means that some of the matrix manipulations can result in dense matrices, and hence do not benefit fully from the special matrix multiplication strategies that we considered for the OLS and WLS_v approaches.

The impact of imposing the non-negativity constraints on the estimation procedure is evaluated by comparing the non-negative reconciled forecasts obtained using each reconciliation approach with the negative reconciled forecasts. Table 7 represents the percentage dif-

ferences in average RMSEs between the reconciled forecasts with non-negatives and negatives at each level in the grouped structure.

It can be seen that the non-negative reconciled forecasts from the OLS approach showed slight gains over the negative reconciled forecasts at each forecast horizon considered (with rare exceptions). These gains are most pronounced at the disaggregated levels. In addition, WLS_v and MinT also showed gains at the most disaggregated level, though the consideration of aggregated levels sometimes introduced slight losses. As these gains or losses are negligible, it can be argued that the non-negativity constraint does not significantly affect the performances of the forecast reconciliation approaches. However, we should emphasize that it is useful in real applications for making meaningful managerial decisions and policy implementations.

6 Conclusions

We have addressed a limitation of existing forecast reconciliation approaches by proposing least squares reconciliation algorithms that are constrained to give non-negative forecasts. Previous forecast reconciliation solutions can give negative values even when all of the base forecasts are non-negative. This is problematic when the data are inherently non-negative in nature, and decisions based on the forecasts require non-negativity (e.g., in budget allocations).

Our approach results in similar reconciled forecasts to the MinT algorithm (and its variants), except that we impose non-negativity constraints during the estimation procedure. This approach may introduce a little bias into the reconciled forecasts, as we are not imposing the conditions that ensure the unbiasedness of the reconciled forecasts.

We considered three algorithms for solving the NNLS problem, namely the block principal pivoting algorithm, the projected conjugate gradient algorithm and the scaled gradient projection algorithm with Barzilai and Borwein updating rules. It was observed that the scaled gradient projection algorithm simplifies to the standard gradient projection algorithm when it is used to obtain a set of non-negative reconciled forecasts from the OLS approach. Hence, the scaled gradient projection algorithm can be extremely computationally demanding for very large structures.

The computational efficiency of these algorithms was evaluated using a set of Monte Carlo simulation experiments and an empirical application using the Australian domestic tourism data. We considered two choices of the initial solution: base forecasts at the most disaggregated level and the (projected) unconstrained least squares solution. The former choice is not applicable for the block principal pivoting algorithm and that-latter choice only allows the block principal pivoting algorithm to be used for OLS and WLS. A projected unconstrained solution is necessary for the projected conjugate gradient algorithm and the scaled gradient projection algorithm, as they need a feasible initial solution. We observed that the unconstrained least squares solution is a better choice. Moreover, the results demonstrated that the block principal pivoting algorithm outperforms the other algorithms considered. This gain in efficiency can be partially attributed to the alternative representation of the MinT approach. The algorithm propagates by (i) removing any infeasible nodes from the structure and (ii) considering the removed nodes as zero, then performing unconstrained least squares on the reduced structure. Hence, the alternative solution is very beneficial in the second stage of the algorithm.

The empirical application evaluates the impact of imposing the non-negativity constraint on the forecast accuracy, along with that of the unconstrained reconciled forecasts. The results demonstrate that the non-negativity constraints introduce slight gains at the most disaggregated level, but slight losses at the aggregated levels. Although these gains and losses are not substantial, a set of non-negative reconciled forecasts is useful for making meaningful managerial decisions in real applications.

Conflict of interest

The authors declare that they have no conflict of interest.

References

Barzilai, Jonathan and Jonathan M Borwein (1988). "Two-point step size gradient methods". In: *IMA Journal of Numerical Analysis* 8.1, pp. 141–148.

Berry, Michael W. et al. (2007). "Algorithms and applications for approximate nonnegative matrix factorization". In: *Computational Statistics and Data Analysis* 52, pp. 155–173.

Bertero, M et al. (2013). "Scaled gradient projection methods for astronomical imaging". In: *EAS Publications Series* 59, pp. 325–356.

Birgin, Ernesto G, José Mario Martínez, and Marcos Raydan (2003). "Inexact spectral projected gradient methods on convex sets". In: *IMA Journal of Numerical Analysis* 23.4, pp. 539–559.

Bonettini, S, R Zanella, and L Zanni (2009). "A scaled gradient projection method for constrained image deblurring". In: *Inverse Problems* 25.1.

Chen, Donghui and Robert J Plemmons (2009). "Nonnegativity constraints in numerical analysis". In: *The birth of numerical analysis*. Ed. by Adhemar Bultheel and Ronald Cools. New Jersey, NJ: World Scientific Publishing, pp. 109–140.

Figueiredo, M A T, R D Nowak, and S J Wright (2007). "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems". In: *IEEE Journal of Selected Topics in Signal Processing* 1.4, pp. 586–597.

Hyndman, Rob J (2019). *forecast: Forecasting functions for time series and linear models*. R package version 8.5. URL: `https://pkg.robjhyndman/forecast`.

Hyndman, Rob J and Yeasmin Khandakar (2008). "Automatic time series forecasting: The forecast package for R". In: *Journal of Statistical Software* 27.3.

Hyndman, Rob J, Alan J Lee, and Earo Wang (2016). "Fast computation of reconciled forecasts for hierarchical and grouped time series". In: *Computational Statistics and Data Analysis* 97, pp. 16–32.

Hyndman, Rob J et al. (2011). "Optimal combination forecasts for hierarchical time series". In: *Computational Statistics and Data Analysis* 55, pp. 2579–2589. DOI: `10.1016/j.csda.2011.03.006`.

Johnson, Norman L, Samuel Kotz, and N Balakrishnan (1994). *Continuous Univariate Distributions*. Ed. by V. Barnett et al. 2nd ed. Vol. 1. Wiley Series in Probability and Mathematical Statistics. New York, NY: Wiley-Interscience.

Júdice, Joaquim J and Fernanda M Pires (1989). "Bard-type methods for the linear complementarity problem with symmetric positive definite matrices". In: *IMA Journal of Management Mathematics* 2.1, pp. 51–68.

— (1994). "A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems". In: *Computers and Operations Research* 21.5, pp. 587–596.

Karjalainen, Erkki J. and Ulla P. Karjalainen (1991). "Component reconstruction in the primary space of spectra and concentrations. Alternating regression and related direct methods". In: *Analytica Chimica Acta* 250, pp. 169–179.

Kim, Jingu and Haesun Park (2011). "Fast nonnegative matrix factorization: An active-set-like method and comparisons". In: *SIAM Journal on Scientific Computing* 33.6, pp. 3261–3281.

Kostreva, Michael M. (1978). "Block pivot methods for solving the complementarity problem". In: *Linear Algebra and its Applications* 21, pp. 207–215.

Lawson, Charles L and Richard J Hanson (1974). *Solving Least Squares Problems*. New Jersey, NJ: Prentice-Hall.

Microsoft Corporation and Steve Weston (2019a). *doParallel: Foreach parallel adaptor for the 'parallel' package*. R package version 1.0.15. URL: `https://CRAN.R-project.org/package=doParallel`.

— (2019b). *foreach: Provides foreach looping construct*. R package version 1.4.7. URL: `https://CRAN.R-project.org/package=foreach`.

Murty, Katta G (1974). "Note on a Bard-type scheme for solving the complementarity problem". In: *Opsearch* 11.2-3, pp. 123–130.

Nocedal, Jorge and Stephen J Wright (2006). *Numerical Optimization*. Ed. by Thomas V. Mikosch, Sidney I. Resnick, and Stephen M. Robinson. 2nd ed. New York, NY: Springer Science and Business Media.

Turlach, Berwin A. and Andreas Weingessel (2019). *quadprog: Functions to solve quadratic programming problems*. R package version 1.5-7. URL: `https://CRAN.R-project.org/package=quadprog`.

Turlach, Berwin A. and Stephen J. Wright (2015). "Quadratic programming". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 7.2, pp. 153–159.

Wickramasuriya, Shanika L, George Athanasopoulos, and Rob J Hyndman (2019). "Optimal forecast reconciliation of hierarchical and grouped time series through trace minimization". In: *Journal of the American Statistical Association* 114 (526), pp. 804–819.

Compiled pdf

Australian tourism dataset used in the manuscript