DECOMPOSING TIME SERIES DATA VIA MIXED INTEGER PROGRAMMING

A Thesis

by

Şeyma Gözüyılmaz

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
February 2020

DECOMPOSING TIME SERIES DATA VIA MIXED INTEGER PROGRAMMING

Approved by:

Assoc. Prof. M. Erhun Kundakcıoğlu,
Advisor
Department of Industrial Engineering
Özyeğin University

Asst. Prof. Mustafa Gökçe Baydoğan
Department of Industrial Engineering
Boğaziçi University

Asst. Prof. Erinç Albey
Department of Industrial Engineering
Özyeğin University

Date Approved: 13 January 2020

To my family...

ABSTRACT

Decomposing time series into seasonality, trend, and remainder reveals underlying insights to be used in forecasting and anomaly detection. Although there are several decomposition methods, no method guarantees all of the following issues are addressed: i) smoothness of trend and the rigid structure of seasonality, ii) shifts in trend, iii) long seasonality periods, iv) multi-seasonality, and v) robustness on outliers. In this study, we propose a mixed integer programming model to address all of these issues. Experiments on different synthetic problem sets present the effectiveness of the proposed algorithm, providing benchmark results against the robust seasonal trend decomposition algorithm [1].

ÖZETÇE

Zaman serilerini trend, sezonsallık ve arta kalan olarak ayırmak, tahmin yapmada ve anormallik belirlemede kullanılacak temelindeki içgörüleri ortaya çıkarmaktadır. Birçok ayrıştırma yöntemi olmasına rağmen, hiçbir yöntem takip eden konuların hepsini ele alacağını garanti etmemektedir. Bu konular i) trendin düzgünlüğü ve sezonsallığın katı yapısı ii) trend'deki değişimler iii) uzun sezonsallık dönemleri iv) çoklu sezonsallık ve v) uç değerlerdeki gürbüzlüktür. Bu çalışmada, tüm bu konuları ele alabilmek adına bir tam sayı programlama modeli öneriyoruz. Farklı sentetik problem kümeleri üzerinde yapılan deneyler, önerilen algoritmanın etkililiğini ve gürbüz sezonsallık trend ayrıştırma algoritmasına karşılık değerlendirme sonuçlarını ortaya koymaktadır.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere thanks to my family for always believing in me and for their continuous and unconditional love and support.

I would like to express my deep and sincere gratitude to my advisor, Dr. Erhun Kundakcıoğlu, for his patience and endless support while I am pursuing my M.S. degree. It was a great privilege and honor to study under his guidance. I am deeply thankful for his friendship and empathy. I also thank all of the professors in the Department of Industrial Engineering for their contributions in my academic and personal development.

I also would like to express my appreciation to the members of my research group, especially Tonguç Yavuz and Cem Deniz Çağlar Bozkır. I would also like to thank my real friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Last but foremost, I am profoundly grateful to Mustafa Kemal Atatürk, the father of our nation, and his companions, without whom I would not be able to provide a scientific contribution by preparing this work in a free and secular country.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Time series are the data indexed in chronological order [2] (e.g., daily stock market quotations, birthrates, the number of influenza cases observed over some time period). Their records can involve more than one variable termed as multivariate series, whereas values of a single variable in time sequence titled as univariate [3]. Series can be discrete or continuous. In a discrete-time series, observations are measured at discrete points of time, while observations are calculated at every instance of time in a continuous type [2] [4].

The stationary state (i.e., parameters such as mean and variance do not change when shifted in time) is a general assumption underlying many statistical procedures used in time series [4]. However, this is not usually the case with actual data. In real life, the value of an observation in a time series is affected by various forces that form the components of time series. There are four components that can be involved: (i) trend, (ii) cyclical variations, (iii) seasonality, and (iv) remainder. The trend component demonstrates the general tendency of data (i.e., increase, decrease, or stagnate) over a long period of time [3]. Despite the fact that the tendencies may change in different intervals of time, the trend is a smooth and long term tendency. In practice, it is assumed that trend also includes cyclical component exists when data exhibit rises and falls not periodically. Seasonality denotes regular and periodic fluctuations based on the season (e.g., every month, quarter, or year). This component is usually supposed to have a fixed and known period. Finally, the remainder describes random and irregular influences. It represents the residuals of the time series after the other components have been removed [4]. The values of time series are calculated as an addition or multiplication of these components [5]. Thus, decomposition can be a method to deal with time series analysis by breaking observed values down into a set of parts [6] [7].

The main objective of time series analysis is producing accurate forecasts from the analysis of the past values, but time series might have undesirable properties (e.g., nonstationarity, irregular fluctuation, or high levels of noise). In-depth analyses of time series data and accurate forecasting have aroused interest for decades [8]. There exist various forecasting tools that involve sophisticated techniques such as pattern imitation [9]. The major advantage of these tools is that they make a prediction without assuming any particular model or distributional characteristics. Nevertheless, obtaining a precise model for time series data in an informed environment is a challenge. One of the most widely-used approaches is *decomposition* of data. While decomposition is not primarily developed to serve as a prediction tool, the underlying idea behind decomposition is segregating the data into smaller parts for representation and insight. Consequently, the prediction problem is also segregated from residuals, which leads to better prediction accuracy [10].

The extraction of time series components occurs as a problem in many applied sciences, and there exist many decomposition methods. Some examples are seasonal trend decomposition using loess, seasonal trend decomposition based on regression, or robust seasonal trend decomposition [10] [5] [1]. Despite the fact that lots of decomposition algorithms and models are available, there are many time-series characteristics that are not addressed in these approaches altogether. The major deficiencies of the main decomposition methods are as follows: i) maintaining the smoothness of trend and periodical structure of seasonality, ii) ability to handle rapid changes in trend, iii) applicability on time series with long seasonality period, iv) capability to overcome multi-seasonality and v) robustness on data with anomalies.

Wen et al. [1] propose a robust technique to overcome these deficiencies. They experiment on varied synthetic and real-time series to demonstrate that their method outperforms existing solutions until that day. Seasonal signals are created by using a square wave with minor random seasonal shifts in the horizontal axis. For this, its period is taken as 50 and 15 seasonal periods are generated. They add a trend signal that comprises of 10 random abrupt changes. In addition, there are 14 spikes and dips as anomalies in the data. The noise is added by zero-mean Gaussian with 0.1 variances. Figure 1 shows generated synthetic data, where the top subplot represents the raw data and trend, the middle subplot represents seasonality, and the bottom subplot represents noise and anomalies. Figure 2 displays the decomposition result that consists of trend, seasonality, and remainder, respectively, by applying the proposed algorithm in [1].

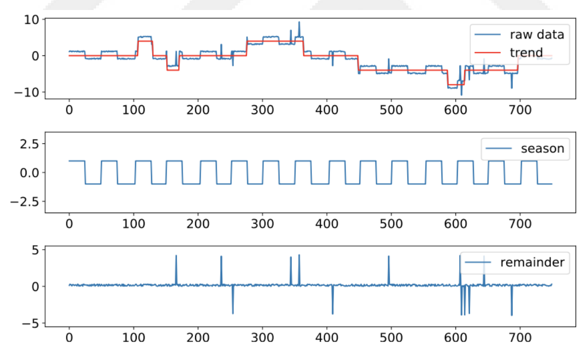

Figure 1: Generated Synthetic Data

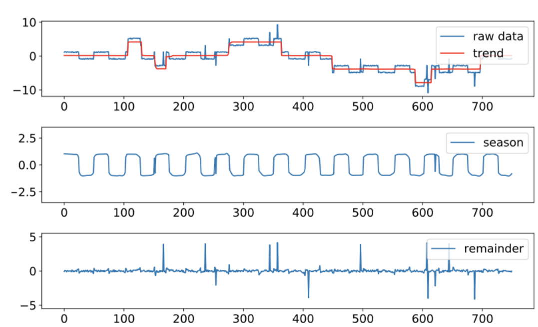

Figure 2: Decomposition Result Using RobustSTL

Considering that all components are very close to the original synthetic signals, we can say that the proposed RobustSTL algorithm is able to separate data into trend, seasonality, and remainder components. From the point of seasonality, this closeness is impressive. It has a fixed and known pattern during a period supposedly. This makes accurate forecasting easier. On the contrary, the structures of the obtained trend and the raw data are approximately the same. This means that the presented algorithm catches most of the rapid changes in trend, but predicting the future is really difficult with that trend distribution. One of the major challenges in time series analysis is getting reasonably accurate forecasts, so this method is not appropriate from the point of prediction with decomposition techniques.

In contrast to all decomposition techniques, mathematical modeling is the tool for exploring natural and human-created phenomena. The mathematical model gives precise and explicit relationships between a set of components and conclusions [11]. Besides this, problems in real life can be accurately represented by mathematical modeling. Large and complex problems can be solved in a timely fashion with this way. The purpose of this study is providing an integer programming model to breakdown time series data into components. At decomposition juncture, our main contribution is filling the gap in the literature by considering all deficiencies mentioned above with the contribution of mathematical optimization. Even though this makes the computational problem potentially more challenging, our study is the first study that reveals the success of mathematical optimization for time-series analysis.

From a different standpoint, no matter what time is the best for any business, it is valuable to be proactive about understanding seasonality and trend in order to plan and prepare strategies. The more that executives can align the business to preferences of customers occur, they are more likely to benefit from choices. Therefore, it is necessary to consider ahead of time of events that are going to grab the attention of customers over the coming times. Therefore, time series decomposition and forecasting procedures are widely used in business [8]. Within this context, the result of our proposed algorithm presents findings that every business leader needs to understand any type of time-series data (e.g., sales, operations, or finance) to overcome these challenges. In addition, our study help managers to make decisions faster and more accurately because of the modeling structure. The remainder of the study is organized as follows: Related literature is reviewed in Chapter 2. In Chapter 3, we formulate a mathematical model that provides the optimal decomposition of time series. Computational results and discussions are presented in Chapter 4. Chapter 5 concludes the study with outcomes and possible future extensions.

CHAPTER II

LITERATURE REVIEW

A growing number of the devices connected to the internet generate a huge volume of data that consists of a sequential set of points listed in time order [2] [12]. Fitting a representative model to this type of data points, aiming accurate forecasts, is termed as time series analysis [4] [3]. Due to the indispensable importance of time series analysis in various fields, numerous methods have been developed by researchers [3]. Decomposition is the prominent method that has been applied among the existing ones. The reason is that time series is driven by multiple factors (e.g., seasonality and trend), and segregation of time series into these feature-based subseries can enhance the forecasting performance of the models [10] [13]. Decomposition techniques were initially developed by Persons (1919) to isolate and identify the salient features of a time series [10]. From this study, many decomposition models are proposed that are different in regards to the main techniques used and abilities. The most known decomposition methods can be classified into groups depending upon these. Thus, the structure of the literature review is as follows:

Moving average based techniques are one of the simplest statistical models available for decomposition. The filtered value is the averaged result within a predefined time period in these methods [14]. One of the most preferred techniques is Seasonal Trend Decomposition Based on Loess Smoothing (STL) within this framework. This methodology is a non-parametric and straightforward way that produces robust outputs from the data which may have missing values. In addition, the period of the seasonal component can be specified as an integer multiple of time sampling interval greater than one [15]. These are the abilities that make the STL more flexible with regards to a better fit to the data. Despite all these facts, this technique often fails to extract the seasonality component accurately when seasonality shift and fluctuation exist [1].

Besides STL, varieties of the auto-regressive integrated moving average (e.g., X-11-ARIMA, X-12-ARIMA, X-13-ARIMA-SEATS, or Seasonal Auto-Regressive Integrated Moving Average) and Seasonal Adjustment at Bell Laboratories (SABL) are some examples of moving average based techniques. The techniques which are extensions of the auto-regressive integrated moving average (ARIMA) contain factors such as external regressors or calendar effects [5]. In addition to this, these algorithms are widely applied in different fields [16]. However, they just handle monthly and quarterly data (i.e., small or medium-size data) [1]. In addition, the differencing processing existing in these methods is not always satisfactory to handle nonlinearity and long seasonality simultaneously [16]. These factors make them inadequate in many aspects, especially when the seasonality period is long. SABL is different in some aspects used to carry out and analyze the decomposition from them [17]. The main characteristic of this method is providing a graphical methodology to assess the performance and behavior of the decomposition [17]. However, this approach never became popular [18].

The second group is subspace-based decomposition techniques. Series are decomposed by taking into account the advantages of subspace learning (i.e., extracting stochastic influences by reducing the dimension of time series) [19]. The Singular Spectrum Analysis (SSA) is an example of subspace-based methods that perform well on short time series [1]. It assumes that series are linear and stationary, and any additional noise follows a normal distribution [19]. This strong assumption makes SSA not applicable to some datasets belong to real cases [1].

Decomposition with multiple regression-based methods is another opportunity for time series analysis. The Hodrick-Prescott filter is one of the basic ways within this context. It works well with slow-changing trend and fast-changing residual [1]. This filter cannot catch up with the abrupt trend changes because of its structure [1]. Seasonal-Trend Decomposition Procedure based on Regression (STR) is proposed depending upon the Hodrick-Prescott filter. STR is flexible to seasonal shifts and multiple seasonalities. In addition to these, it provides confidence intervals for the predicted components [1] [5]. Robust version is proposed to deal with spikes and dips better, and consequently, it works well with outliers [1]. However, both STR and robust STR cannot follow abrupt change on the trend [1].

Transformation based methods are strategies to extract stochastic and deterministic influences from time series [19]. Fourier and Wavelet transforms are the most known examples. Fourier Transform (FT) breaks down a time series into a sum of sinusoidal functions [19]. This technique is applicable in several fields where deterministic and stochastic parts occur at different frequency intervals [19]. Nevertheless, it faces issues when analyzing non-periodic and non-stationary data [19]. The Wavelet Transform (WT) was provided to develop the results from FT. It is performed on different scales to extract components at different frequencies [19]. Although Wavelet transform presents additional properties to Fourier transform, it jeopardizes the time relationship among observations due to the way of its work [19], which affects modeling and forecasting.

CHAPTER III

MATHEMATICAL MODEL

Decomposition is one of the techniques that are widely used to perform time series analysis. In this section, the time series is subdivided into three additive components (i.e., trend, seasonality, and remainder) through a mathematical model. The behaviors of times series differ from one problem set to another, but there are general expectations to produce accurate forecasts. In this context, seasonality typically describes regular and periodic patterns. It is supposed to have a fixed and known period. The trend component demonstrates the continuous tendency of data. Finally, the remainder involves all values other than trend and seasonality.

We build our basic model based on these expectations and we use the following notation:

Sets:

T : set of time points,

R : set of time points in a season that recurs.

In our model, parameters are represented by the followings:

Parameters:

ρ : allowable percentage for change between two consecutive trend values,

Δ : difference between the max and min observed value,

B : upper bound on instance number of trend shifts,

K : upper bound on instance number of anomalies (i.e., spike and dip movements),

X_t : value of data point at time t.

X_t denotes the data instances at time t where t $\in T$. Our purpose is to decompose these instances into seasonality, trend, and remainder components via mixed integer programming model. Before we proceed with the model, we define the many-to-one matching between the set of time points and the set of time points in a season that recurs. Let $f : T \rightarrow R$ be the function defined by the following equation:

$$f(t) = \{t(mod|R|)\}, t \in T \tag{1}$$

The function above is identified to represent the time points in T that have the same seasonality values. Let the length of a season (i.e., $|R|$) is 20 to make this clear (e.g., $t = 20$, $t = 40$, $t = 60$ etc). Based on this assumption, the seasonality at time t should get same value in every iterated 20 time points. The modulo operation in the function serves this purpose.

We set our basic mathematical model with the following decision variables:

Decision Variables:

X_t^{Trend} : trend value at time t, $t \in T$

X_r^{Season} : seasonality value at time r, $r \in R$

$X_t^{\text{Remainder}}$: remainder value at time t, $t \in T$

As mentioned above, the aim of this study is to break the observed value at time t down into a set of its components. The base model is formulated as follows:

$$\min \sum_{t \in T} (X_t^{\text{Remainder}})^2 \tag{2a}$$

$$\text{s.t.} \quad X_t = X_t^{\text{Trend}} + X_r^{\text{Season}} + X_t^{\text{Remainder}}, \quad \forall t \in T, r = f(t) \tag{2b}$$

$$\sum_{r \in R} X_r^{\text{Season}} = 0 \qquad \forall r \in R \tag{2c}$$

In this formulation, the objective function is to minimize the sum of square of all remainder values in T. It is represented like this due to the fact that the remainder value can be negative. There are two models generally used for time series: (i) additive model and (ii) multiplicative model. Constraint (2b) ensures that the additive model is used in this mathematical formulation. Constraint (2c) preserves the rigid structure of seasonality because seasonality variables can take negative values.

$$X_t^{\text{Trend}} - \rho\Delta \leq X_{t+1}^{\text{Trend}} \leq X_t^{\text{Trend}} + \rho\Delta, \qquad \forall t \in T \tag{3a}$$

Constraint (3a) ensures that the trend is smooth. In the stable cases (i.e., no shifts in trend), the trend value from time t to $t+1$ can vary up to allowable percentage (ρ) of the difference between the maximum and the minimum observed value (Δ) in time series.

In the next subsections, we present how certain issues faced can be handled, extending the base model.

3.1 Shifts in Trend

The general tendency of points over a long period of time is denoted by the trend component. Despite the fact that the tendencies may change in time, it is assumed that the trend is smooth to make accurate predictions. However, abrupt shifts in trend may occur in real cases. To handle these changes and provide smoothness, the following decision variables are presented:

New Decision Variable and Parameter:

Y_t : 1 if trend has an abrupt change (i.e., a spike or dip) at time t, 0 otherwise,

M : large enough number.

The based model is extended by updating the constraint (3a), and adding the new constraint (i.e., (4c)):

$$X_{t+1}^{\text{Trend}} \leq X_t^{\text{Trend}} + \rho\Delta + MY_t, \qquad \forall t \in T \tag{4a}$$

$$X_{t+1}^{\text{Trend}} \geq X_t^{\text{Trend}} - \rho\Delta - MY_t, \qquad \forall t \in T \tag{4b}$$

$$\sum_{t \in T} Y_t \leq B \tag{4c}$$

$$Y_t \in \{0, 1\} \qquad \forall t \in T \tag{4d}$$

The binary decision variable Y_t states that if the trend has an abrupt change or not at time t. In addition, the user-defined parameter B denotes the maximum number of spike and dip movements. Constraint (4c) ensures that the allowed number of spike and dip movements cannot be more than B. Besides this condition, constraints (4a) and (4b) provide that the value of a trend at time $t+1$ should remain between certain ranges. On the contrary, the trend at time $t+1$ can take any value by courtesy of a sufficiently large constant (M). This constant is considered as the difference between the fifth maximum and fifth minimum of observed values in the proposed mixed integer model. Before this, we assigned a haphazard (M) value, which produced loose bounds. Therefore, we changed it by taking trend shifts into consideration.

3.2 Multi-Seasonality

Within the concept of seasonality, it is considered as a regular and periodic pattern. Nevertheless, time series representing real-world activity are often affected by several schedules representing several seasonal periods. This circumstance can be solved with the contributions of the following changes:

First of all, the new set and index are added to the base model. With reference to these, R and r are altered.

New Sets:

J : set of seasons,

R_j : set of time points in a season j that recurs, $j \in J$.

Next, the length of a season type j is added as a parameter to the base model.

New Parameter:

$|R_j|$: length of a season type j.

In addition, the many-to-one matching represented by function f in (1) is updated as follows:

$$f_j(t) = \{t(mod|R_j|)\}, t \in T, j \in J \tag{5}$$

Finally, the formulation of the base model is updated by adding a new index j to the current decision variable X_r^{Season} and changing constraints (2b) and (2c). Their new forms are as follows:

Decision variables:

$X_{r,j}^{\text{Season}}$: seasonality value of seasonality type j at time r, $r \in R_j$, $j \in J$.

Constraint 2b *and* 2c *updated versions:*

$$X_t = X_t^{\text{Trend}} + \sum_{j \in J} X_{f_y(t),j}^{\text{Season}} + X_t^{\text{Remainder}}, \forall t \in T \tag{6}$$

$$\sum_{r \in R_j} X_{r,j}^{\text{Season}} = 0, \forall j \in J \tag{7}$$

3.3 Unexpected Instances

In a traditional decomposition procedure, seasonality denotes regular fluctuations based on the season and trend is smooth. However, unexpected cases may occur in reality (e.g., the value of seasonality may change in one of the seasons). The based model above can be extended to address this deficiency by adding new decision variable:

New Decision Variable:

$X_t^{\text{Remainder'}}$: remainder prime value at time t, $t \in T$,

Z_t : 1 if there is an unexpected instance at time t, 0 otherwise.

Second, the based model is extended by adding the following constraints:

$$X_t^{\text{Remainder'}} \leq X_t^{\text{Remainder}} + MZ_t, \qquad \forall t \tag{8a}$$

$$X_t^{\text{Remainder'}} \geq X_t^{\text{Remainder}} - MZ_t, \qquad \forall t \tag{8b}$$

$$\sum_{t \in T} Z_t \leq K \tag{8c}$$

Finally, the objective function of the base model is updated by changing $X_t^{\text{Remainder}}$ to $X_t^{\text{Remainder'}}$. New form of the objective function is the following:

$$\min \quad \sum_{t \in T} (X_t^{\text{Remainder'}})^2 \tag{9a}$$

CHAPTER IV

COMPUTATIONAL RESULTS

We performed experiments to present the effectiveness of our proposed algorithm on synthetic problem sets. In these sets, just a single season is considered because the RobustSTL algorithm does not present the cases with multi-seasonality. Despite this, some time series with multiple seasons are generated to give an idea. The robust seasonal trend decomposition algorithm is used as a baseline for comparison purposes. The reason is that they demonstrated the efficiency and practical usefulness of their algorithm by comparing it with the three state-of-the-art baseline algorithms [1].

4.1 Data Description

Our synthetic data is the additive variation of the components that involve seasonality, trend, and remainder. All these components are generated depending upon different assumptions as noted below:

Seasonality

Seasonality values are generated from a Bernoulli distribution and cosine function in the case of single seasonality. The Bernoulli distribution is a discrete distribution having two possible outcomes labeled by 0 and 1. Differently from this, outcomes are -1 and 1 based on our assumption. The cosine function (f(x) = $cos(x)$) has all real numbers in its domain but its range is -1 $\leq cos(x) \leq$ 1. Our seasonality values change in this domain.

Trend

This component continuously increases or decreases or moves randomly. Its values are generated depending upon different rules for each situation. First of all, the natural logarithm functions $ln(x)$ is used to generate an increasing trend. The output for ln is unrestricted, and its domain is infinity from both positive and negative sides. We take into consideration just the positive side. Furthermore, a linear function is used to represent a decreasing trend. In both cases, rates of increase and decrease are different but 10 or 100 is set as initial values of all trend series. Finally, trend shifts within the scope of randomness. As part of some problem sets, its value changes by 2 during the shift. On the contrary, trend value alters by 3 in some other sets.

Remainder

Remainder values are generated with a normal distribution. This distribution is a type of continuous probability distribution for a real-valued random variable. It takes two parameters that are mean (i.e., expectation of the distribution) and standard deviation. In our problem sets, the mean is 0, and the standard deviation is 0.5. Remainder values change dramatically at random points. The rate of these abrupt changes can be 5, 7.5, or 10.

In total, 60 synthetic data sets are generated by considering trend shifts, multi-seasonality, and unexpected instances to simulate real-world scenarios. These problem sets can be grouped based on the number of data points and the length of a season. Five experimental groups can be derived from this point of view: i) short time series with a short season, ii) short time series with a medium season, iii) long time series with a short season, iv) long time series with a medium season, v) long time series with a long season. The long seasonality period is just considered for long data. The reason is that the number of seasons is not enough to represent the existence of seasonality with the assumptions that we make.

4.2 Parameter Setting

The proposed mixed integer model uses six different parameters related to number of data points, length of a season, upper bounds, change between two consecutive trend values and observed values. In spite of this, the robust seasonal trend decomposition algorithm has the capability to handle the decomposition of time series with less parameters. The RobustSTL algorithm gains an advantage over the proposed algorithm with this ability. During the experiments, the following parameters are used for each algorithm:

Table 1: Parameter Comparison of Algorithms

Parameters	Mixed Integer Programming	RobustSTL
Number of Data Points ($\|T\|$)	•	•
Length of a Season ($\|R\|$)	•	•
Upper Bound on Instance Number of Trend Shifts (B)	•	
Upper Bound on Instance Number of Anomalies (K)	•	
Allowable Percentage for Change Between Two Consecutive Trend Values (ρ)	•	
Difference Between the Max and Min Observed Value (Δ)	•	

The number of data points ($|T|$) is accepted as 100 and 400 for the problem sets. The length of a season ($|R|$) is considered as 10, 20, or 50. The upper bound on instance number of trend shifts (B) is considered as 10 for all generated problems with random trend shifts. This value is accepted as 0 when the trend increases or decreases continuously. Based on the number of data points, the upper bound on instance number of anomalies (K) changes. If the data set includes 100 time points, anomalies may happen at 10 points. Conversely, when the number of data points is higher (i.e., 400), unexpected values occur at 15 points.

Beside these, the allowable percentage for change between two consecutive trend values (ρ) is taken as (1 / $|T|$) for all test problems. Finally, the difference between the maximum and minimum observed value (Δ) in data set depends on problem set. It is considered as the difference between the fifth maximum and fifth minimum observed value. The reason is that there are trend shifts and unexpected abrupt. Therefore, the maximum and minimum values of problem set can be occurred at these points and this may mislead the results. We considered the fifth maximum and minimum values to define (Δ) to handle this possibility. The following table summarizes all these values with regards to five experimental groups as mentioned at the end of the data description part:

Table 2: Parameter Summary

Parameters	Group 1 Short Series Short Season	Group 2 Short Series Medium Season	Group 3 Long Series Short Season	Group 4 Long Series Medium Season	Group 5 Long Series Long Season
$\|T\|$	100	100	400	400	400
$\|R\|$	10	20	10	20	50
B	0 or 10	0 or 10	0 or 10	0 or 10	0 or 10
K	10	10	15	15	15
ρ	1 / $\|T\|$	1 / $\|T\|$	1 / $\|T\|$	1 / $\|T\|$	1 / $\|T\|$

4.3 Decomposition Results

Decomposition results via the proposed mixed integer programming is presented as figures to shed light on the visualization of numerical data. Subplots display the raw data, seasonality, trend, and remainder parts, respectively, in each figure. The seasonality component comprises of two different seasonality in multi-seasonality figures. Therefore, figures in the multi-seasonality part show decomposed data where the top subplots represent the raw data and trend, the first and second subplots represent seasonality 1 and seasonality 2, and finally, the bottom subplots represent noise and anomalies.

The solid and bold black lines represent the decomposition values using the proposed MIP model, where the gray and dashed lines indicate generated data for each component. The term MIP will be used in the rest of the thesis. Actually, minimizing the summation of square of errors is one of the most widely used approaches for the objective of similar problems. When we introduce the binary variables, the problem becomes a mixed integer nonlinear programming (MINLP) problem. Since the nonlinear term is fashionably in the objective function only and we propose adding integer (i.e., binary) variables, we use MIP for brevity. The following figures are provided for each issue (i.e., shifts in trend, multi-seasonality, and unexpected instances) handled by the proposed mixed integer programming model:

Shifts in Trend

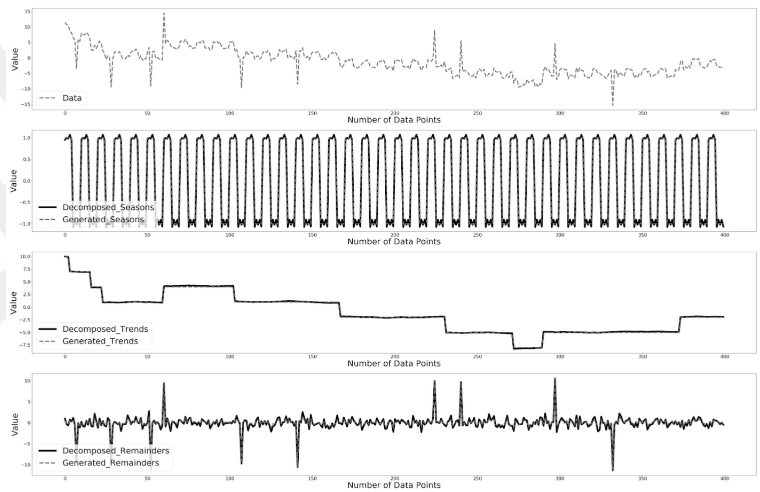

Figure 3: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

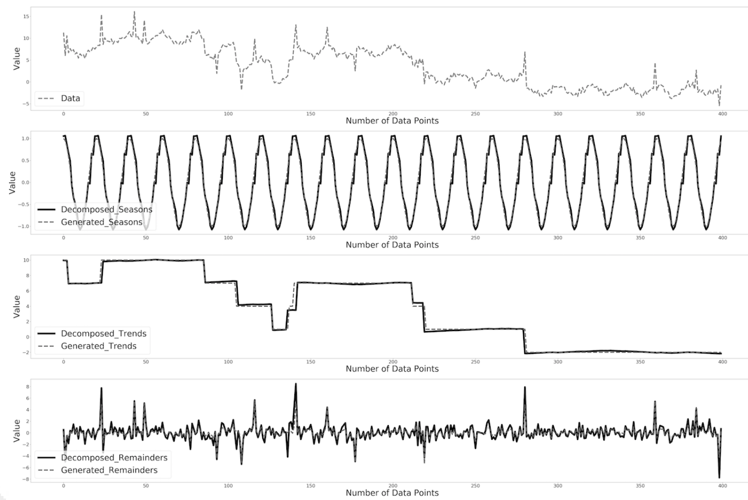

Figure 4: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

Multi-Seasonality

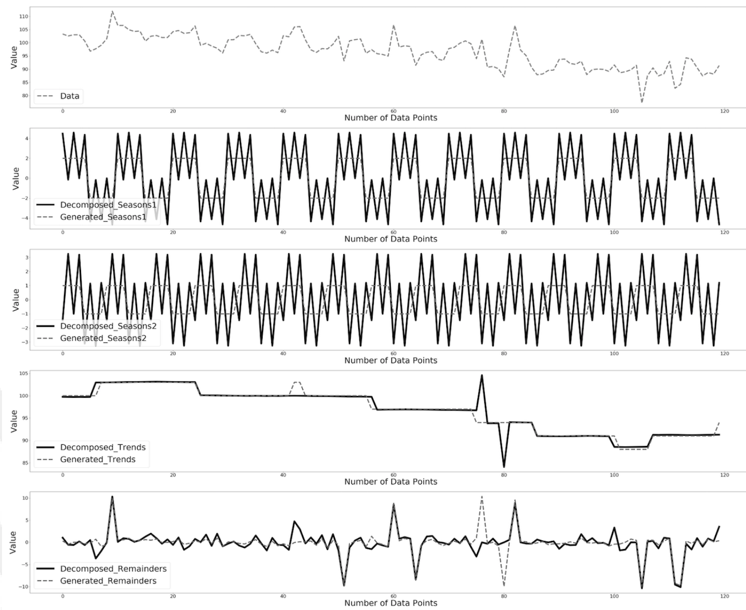

Figure 5: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

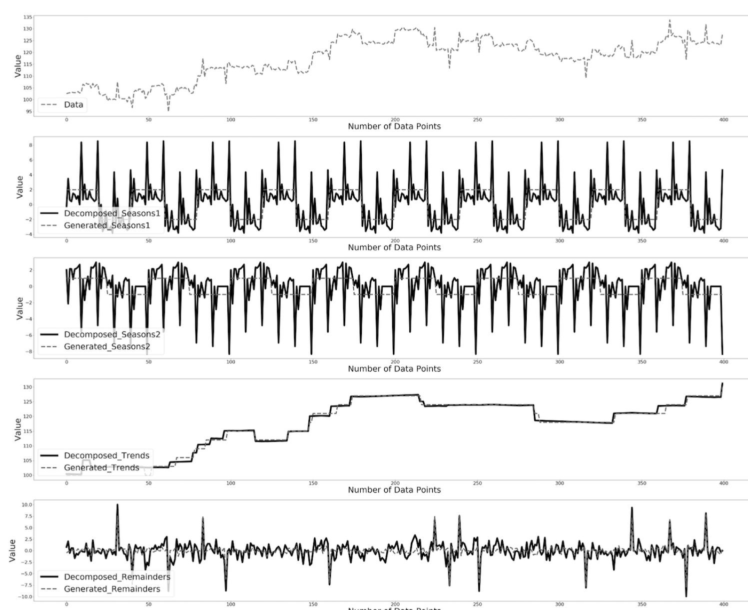

Figure 6: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

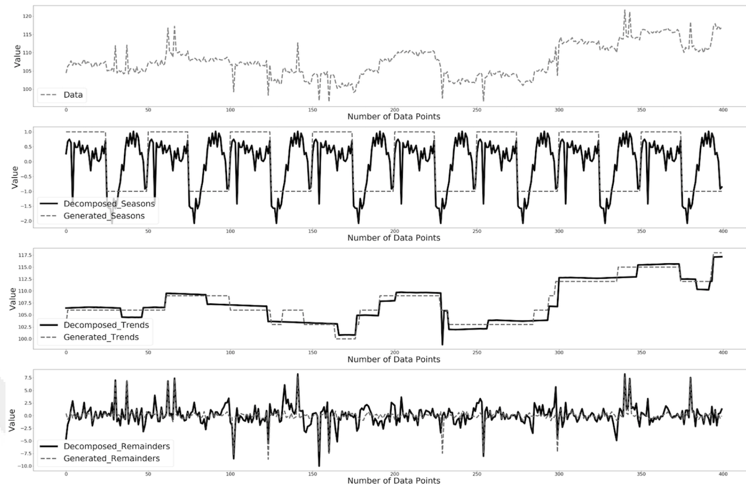

Figure 7: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

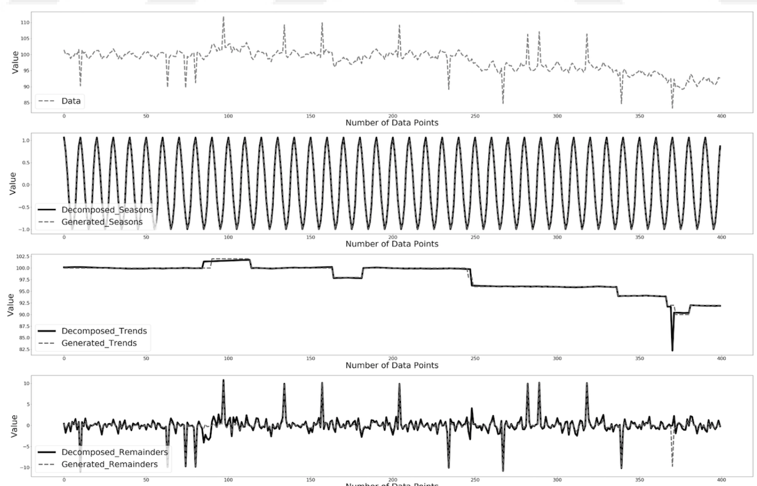

Figure 8: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

4.4 Methods Comparison

The robust seasonal trend decomposition algorithm [1] is introduced as the benchmark for comparison to validate the optimization ability of the proposed mixed integer programming model. They use three state-of-the-art baseline algorithms for comparison purposes that are seasonal trend decomposition using loess, trigonometric exponential smoothing state-space model with box-cox transformation and seasonal trend decomposition procedure based on regression in [1]. Consequently, we take many algorithms into consideration by comparing our algorithm to the robust seasonal trend decomposition.

Box plots are used to detect the differences between the results of both algorithms on average. These differences are demonstrated by the distribution of data depending on a five-number summary (i.e., minimum, first quartile, median, third quartile, and maximum).

Box plots are presented under different subtitles for each component of time series that are remainder, seasonality, and trend, respectively. Data in the figures are classified according to five experimental groups: i) short time series with a short season, ii) short time series with a medium season, iii) long time series with a short season, iv) long time series with a medium season, v) long time series with a long season.

The problem set is termed as short and long data when it involves 100 and 400 time points, respectively. Season type is classified depending on the length of a season. The length is considered as 10, 20, and 50 that indicate short, medium, and long season types, respectively. They are represented with the letters "S", "M" and "L" in horizontal axes of figures. Long season is just taken into consideration for long data. This is because the number of replications is not enough to represent the existence of seasonality with the assumptions that we make. In this study, the trend can move randomly or continuously increase or decrease. For each case of the trend, figures are presented in the following order under subtitles: random trend, increasing trend, and decreasing trend. Trend series start from 10 in the first two box plots in each row of all figures. 100 is an initial value of trend series in the second two box plots in each row of all figures. The first row of each figure includes the data sets whose seasonality values are generated from a Bernoulli distribution. Seasonality values of the problem sets in the second row of each figure are generated via cosine function.

Box Plots for Remainder Component

Figures 5,6 and 7 display the differences between the generated and decomposed values of the remainder component for both algorithms in the cases of trend with shifts, continuously increasing and decreasing trend, respectively.

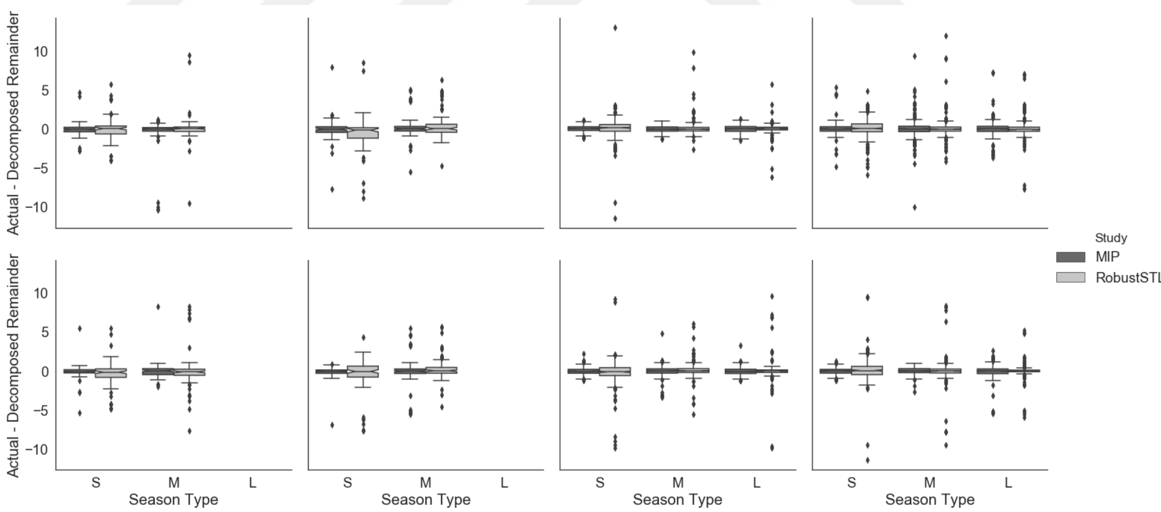

Figure 9: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

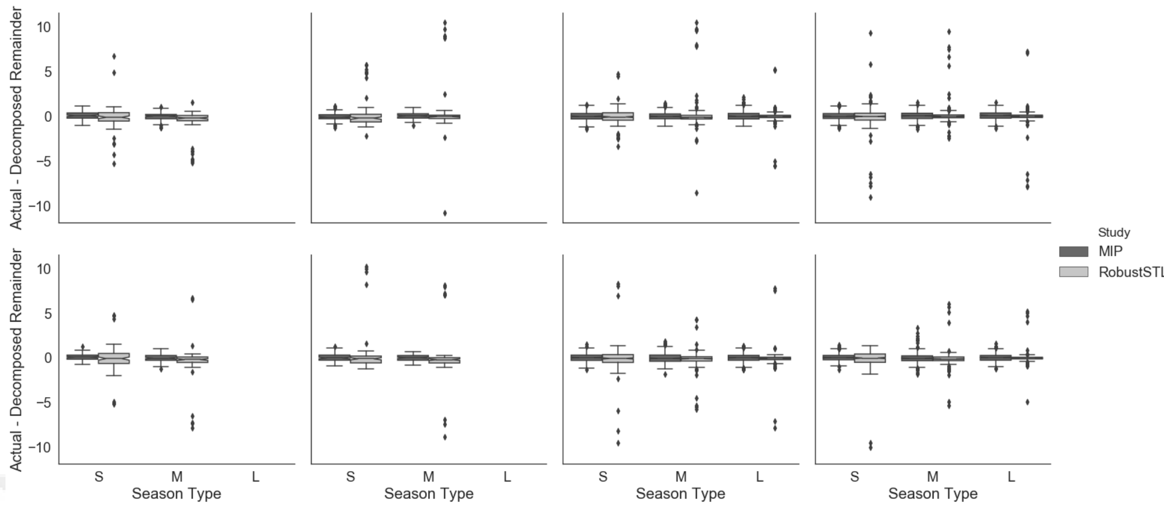

Figure 10: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

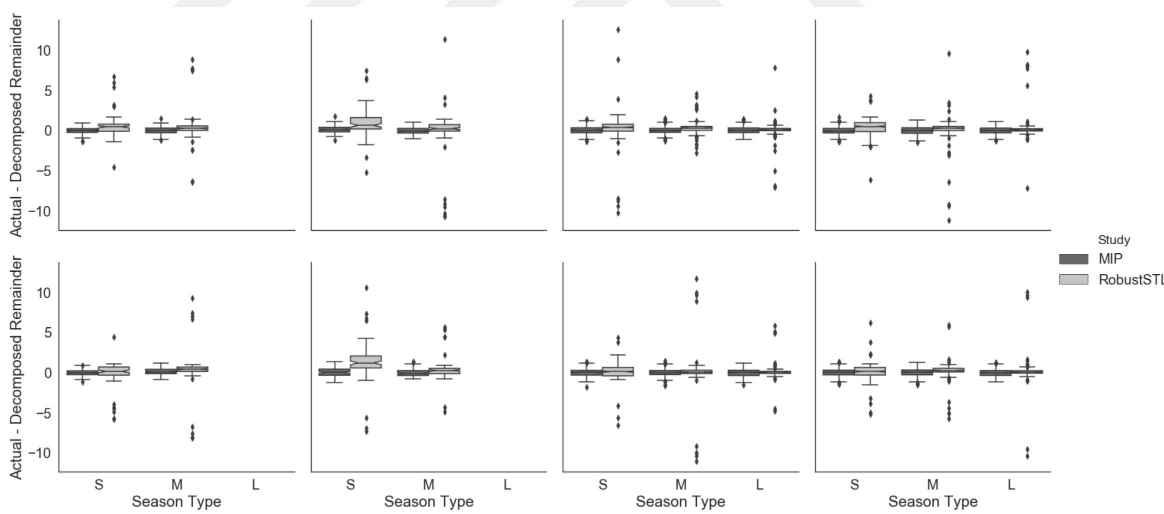

Figure 11: Actual-Decomposed Remainder Comparison for RobustSTL & Our Proposed MIP

It can be understood from the figures above that the decomposed values of the proposed mixed integer algorithm and the robust seasonal trend decomposition algorithm are close to each other in many experimental groups. The proposed algorithm shows better decomposition performance in several cases. In most of the cases with close results, the robust seasonal trend decomposition algorithm has outliers with higher values, which may be evaluated as a negative point.

Box Plots for Seasonality Component

Figures 8,9 and 10 show the performance assessments of the MIP and RobustSTL for the differences between the generated values and decomposed values of the seasonality component in the cases of trend with shifts, continuously increasing and decreasing trend, respectively.

From the figures below, it is clear that the decomposition performance of the proposed mixed integer programming model is better than the robust seasonal trend decomposition algorithm in most of the cases. This proves the effectiveness of the mixed integer model for improving the performance of RobustSTL in regards to the seasonality component.

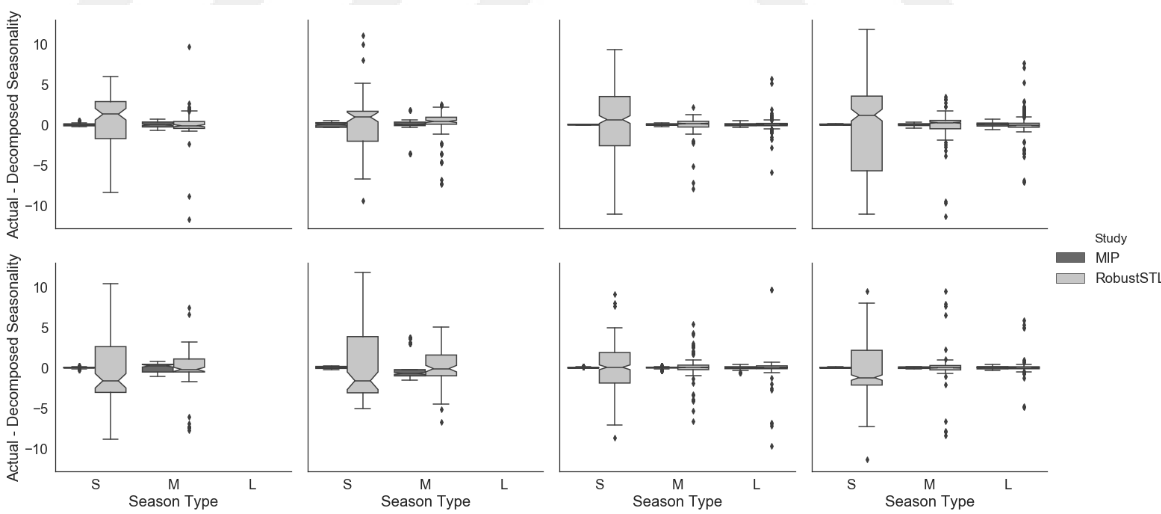

Figure 12: Actual-Decomposed Seasonality Comparison for RobustSTL & Our Proposed MIP

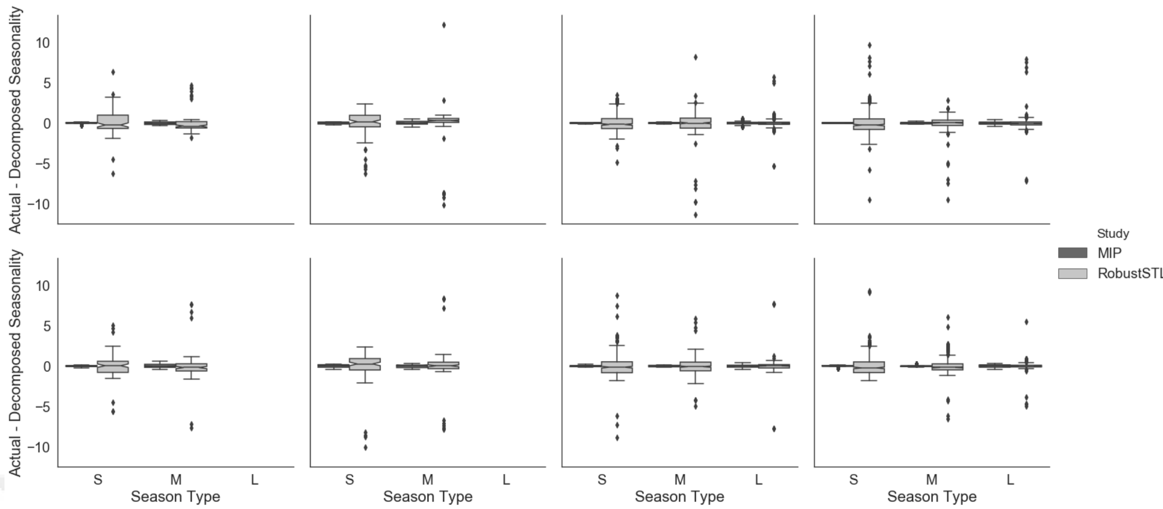

Figure 13: Actual-Decomposed Seasonality Comparison for RobustSTL & Our Proposed MIP

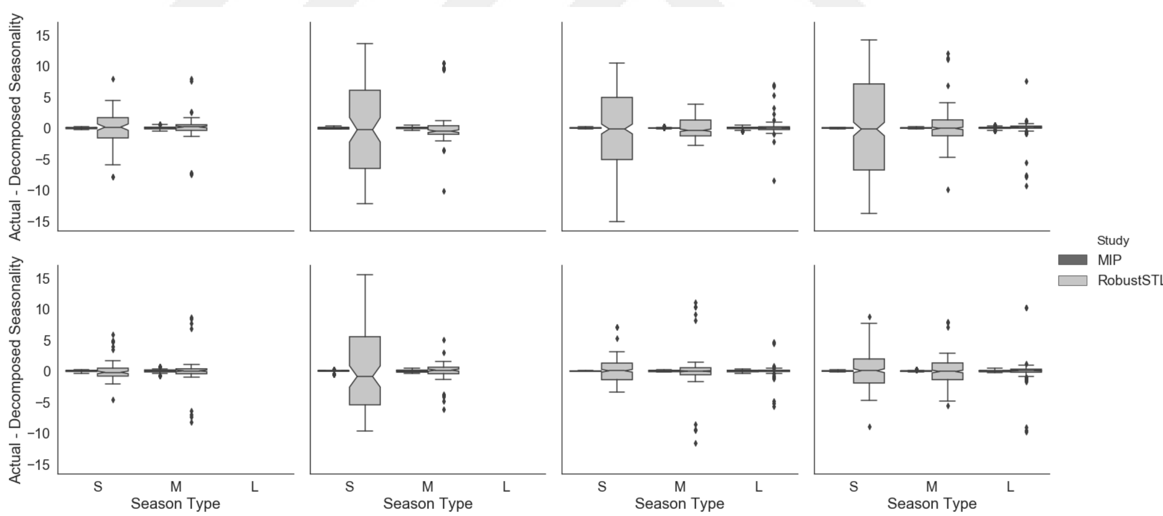

Figure 14: Actual-Decomposed Seasonality Comparison for RobustSTL & Our Proposed MIP

Box Plots for Trend Component

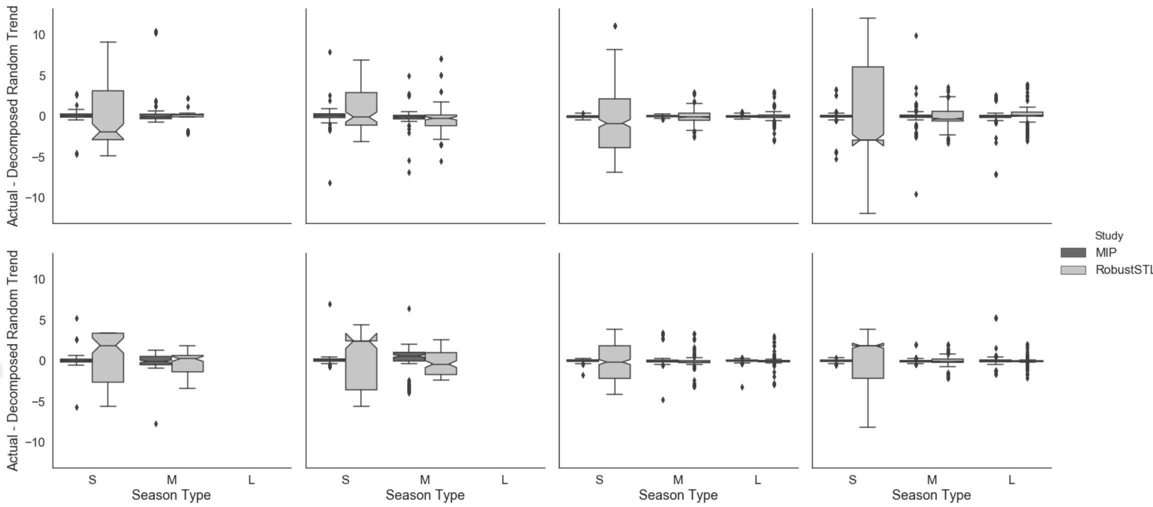

Figure 15: Actual-Decomposed Trend Comparison for RobustSTL & Our Proposed MIP

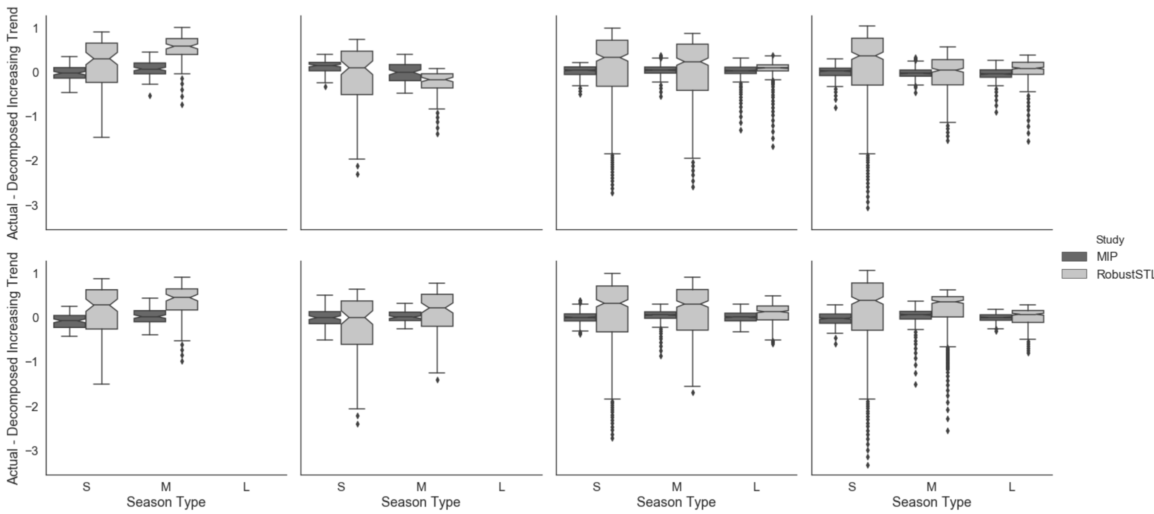

Figure 16: Actual-Decomposed Trend Comparison for RobustSTL & Our Proposed MIP

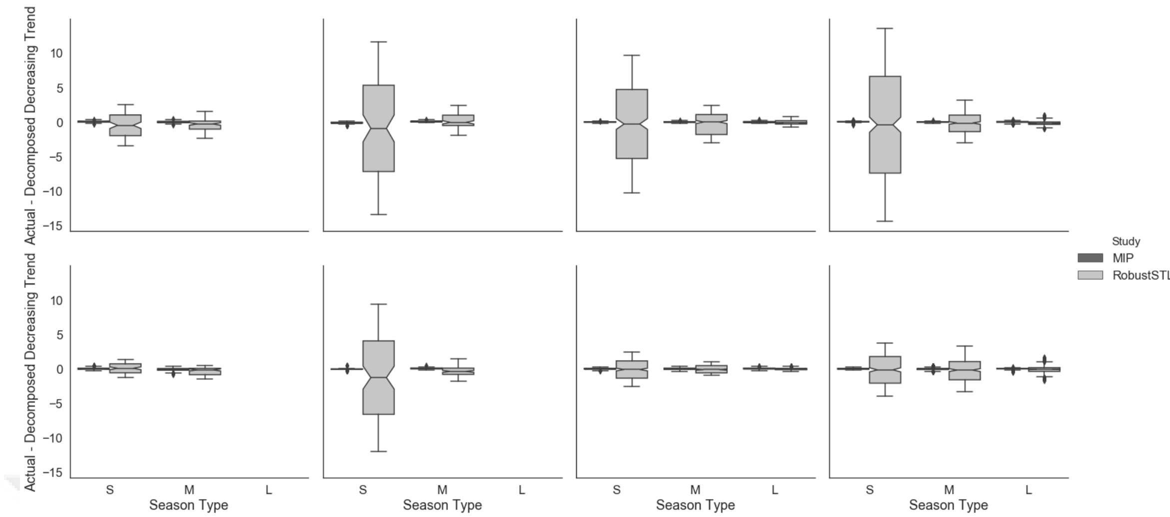

Figure 17: Actual-Decomposed Trend Comparison for RobustSTL & Our Proposed MIP

Figures 11, 12, and 13 above show the differences between the generated and decomposed values of the trend part for both algorithms in the cases of trend with shifts, continuously increasing and decreasing trend, respectively. It is clear that the decomposition performance of the proposed mixed integer programming model indeed enhances compared to the robust seasonal trend decomposition algorithm, especially for trend component. This outcome offers an insight into the applicability of the proposed algorithm in this paper.

In addition to these box plots, the following tables are used to present the variance values of the difference between the generated data and decomposed data. The values are presented for both algorithms.

Variance Table for Remainder Component

	Shifts in Trend		**Increasing Trend**		**Decreasing Trend**	
	MIP	**RobustSTL**	**MIP**	**RobustSTL**	**MIP**	**RobustSTL**
Instance1	**0.77**	2.319	**0.185**	1.720	**0.171**	1.678
Instance2	4.087	**2.933**	**0.195**	1.531	**0.229**	3.582
Instance3	**1.731**	4.707	**0.197**	2.164	**0.199**	2.495
Instance4	**1.716**	2.584	**0.150**	5.742	**0.171**	7.736
Instance5	**0.149**	1.796	**0.227**	0.450	**0.215**	1.815
Instance6	**0.166**	0.713	**0.198**	1.657	**0.207**	0.339
Instance7	**0.203**	0.441	**0.265**	0.421	**0.200**	0.570
Instance8	**0.511**	1.226	**0.225**	1.326	**0.205**	0.622
Instance9	**1.277**	1.415	**0.203**	0.851	**0.238**	1.385
Instance10	**0.771**	1.286	**0.213**	0.863	**0.207**	1.121
Instance11	**0.832**	2.571	**0.145**	1.787	**0.154**	1.964
Instance12	**1.000**	4.935	**0.166**	3.078	**0.201**	4.870
Instance13	**0.595**	4.373	**0.164**	3.875	**0.291**	4.527
Instance14	2.889	**1.808**	**0.128**	4.788	**0.227**	1.827
Instance15	**0.185**	1.953	**0.204**	1.229	**0.214**	0.653
Instance16	**0.367**	0.630	**0.281**	0.510	**0.194**	2.395
Instance17	**0.216**	1.329	**0.193**	0.623	**0.219**	0.360
Instance18	**0.181**	1.713	**0.177**	0.783	**0.210**	0.631
Instance19	**0.187**	1.437	**0.307**	0.527	**0.217**	0.609
Instance20	**0.519**	0.595	**0.190**	0.322	**0.226**	1.273

Variance Table for Seasonality Component

	Shifts in Trend		**Increasing Trend**		**Decreasing Trend**	
	MIP	**RobustSTL**	**MIP**	**RobustSTL**	**MIP**	**RobustSTL**
Instance1	**0.045**	17.380	**0.015**	2.137	**0.020**	5.440
Instance2	**0.146**	3.801	**0.037**	1.549	**0.088**	4.028
Instance3	**0.083**	10.912	**0.014**	2.625	**0.030**	54.480
Instance4	**0.879**	3.480	**0.066**	6.009	**0.048**	8.160
Instance5	**0.004**	15.145	**0.003**	1.227	**0.008**	34.828
Instance6	**0.016**	0.630	**0.007**	2.157	**0.007**	2.447
Instance7	**0.037**	0.383	**0.032**	0.453	**0.034**	0.634
Instance8	**0.004**	31.147	**0.003**	2.125	**0.006**	66.201
Instance9	**0.035**	1.428	**0.013**	0.912	**0.009**	3.611
Instance10	**0.066**	1.373	**0.036**	0.864	**0.031**	1.149
Instance11	**0.013**	13.706	**0.011**	2.296	**0.021**	2.242
Instance12	**0.353**	5.404	**0.068**	3.406	**0.097**	4.943
Instance13	**0.024**	15.477	**0.041**	4.219	**0.036**	42.033
Instance14	**3.044**	3.391	**0.042**	4.930	**0.056**	1.842
Instance15	**0.005**	6.010	**0.012**	1.962	**0.006**	2.846
Instance16	**0.013**	0.851	**0.011**	0.870	**0.017**	2.600
Instance17	**0.040**	1.173	**0.029**	0.670	**0.030**	0.351
Instance18	**0.004**	8.770	**0.010**	1.603	**0.009**	5.877
Instance19	**0.012**	1.384	**0.010**	0.914	**0.008**	3.037
Instance20	**0.039**	0.506	**0.035**	0.332	**0.027**	1.388

Variance Table for Trend Component

	Shifts in Trend		**Increasing Trend**		**Decreasing Trend**	
	MIP	**RobustSTL**	**MIP**	**RobustSTL**	**MIP**	**RobustSTL**
Instance1	**0.667**	19.01	**0.036**	0.382	**0.013**	3.092
Instance2	4.371	**0.630**	**0.027**	0.109	**0.022**	0.732
Instance3	**1.615**	7.273	**0.022**	0.540	**0.015**	53.672
Instance4	**1.380**	3.703	**0.046**	0.084	**0.010**	0.996
Instance5	**0.023**	14.455	**0.013**	0.657	**0.005**	33.584
Instance6	**0.015**	0.449	**0.015**	0.518	**0.008**	2.500
Instance7	**0.023**	0.310	**0.030**	0.050	**0.013**	0.109
Instance8	**0.331**	32.790	**0.018**	0.727	**0.004**	65.825
Instance9	**0.731**	0.831	**0.012**	0.154	**0.007**	2.032
Instance10	**0.449**	1.248	**0.018**	0.059	**0.011**	0.087
Instance11	**0.804**	11.814	**0.030**	0.382	**0.020**	0.581
Instance12	**0.977**	1.358	**0.030**	0.155	**0.053**	0.288
Instance13	**0.533**	13.399	**0.040**	0.540	**0.008**	38.961
Instace14	3.418	**2.115**	**0.016**	0.245	**0.012**	0.542
Instance15	**0.025**	5.310	**0.016**	0.657	**0.011**	2.099
Instance16	**0.234**	0.556	**0.018**	0.309	**0.020**	0.346
Instance17	**0.042**	0.201	**0.015**	0.062	**0.013**	0.025
Instance18	**0.020**	8.701	**0.019**	0.772	**0.008**	4.978
Instance19	**0.039**	0.251	**0.035**	0.293	**0.021**	2.744
Instance20	0.283	**0.113**	**0.008**	0.034	**0.006**	0.241

Finally, from the point of computational time, the robust seasonal trend decomposition algorithm is indeed more time saving compared to the proposed method.

Although these are problems that do not need to be solved in seconds or minutes, especially in the domain of business analytics, a thorough analysis in a couple of hours is quite welcome. Fast tools can be used in a number of domains and to provide input to more sophisticated tools, so they have their uses. However, in-depth analysis that provides further insights is of vital importance, despite potentially longer run times. In this study, problem sets that involve continuously increasing or decreasing trends are solved in less time compared to the data sets with trend shifts.

CHAPTER V

CONCLUSION

Extraction of the time series components could provide a guide to make accurate predictions from the analysis of the past values. It is extremely significant to enhance the forecasting accuracy of time series because they occur in many areas (e.g., yearly sales figures, monthly rainfall data, or hourly readings of air temperature). However, obtaining accurate prediction is a challenging task due to undesirable properties of time series.

Thus, we proposed the mixed integer mathematical model to decompose complex time series into trend, seasonality, and remainder components with the capability to handle the deficiencies. Our proposed algorithm is novel since mathematical modeling has not been applied for time series decomposition in the literature, to the best of our knowledge. The robust seasonal trend decomposition algorithm of [1] is used as a baseline for comparison.

Experimental results on synthetic data sets have demonstrated the effectiveness and the practical usefulness of our algorithm. Decomposition results of the remainder component can be seen as poor performance. Due to the fact that decomposed values of the proposed algorithm and the RobustSTL algorithm are close to each other in many problem sets. However, the robust seasonal trend decomposition algorithm has outliers with higher values. In terms of the seasonality component, the decomposition performance of the proposed mixed integer programming model is better than the robust seasonal trend decomposition algorithm in most of the data sets, especially with the short-season period. The decomposition performance of the proposed model indeed enhances compared to the RobustSTL algorithm for the trend component.

Finally, the RobustSTL algorithm can decompose time series in less time.

As for the future research directions, first, new mathematical models can be developed, containing related formulas to find the best allowable percentage for change between two consecutive trend values, best upper bound on instance number of trend shifts, best upper bound on instance number of anomalies and M large enough number. This will make the model more realistic in terms of assumptions and more comparable with regards to the number of parameters used. Second, the decomposition results of the RobustSTL algorithm can be considered as a warm start for the proposed model. This may shorten the computation time of the MIP. In addition, the expected time of the anomalies can be provided to the proposed algorithm to enhance the quality of the decomposition and shorten the solution time. Lower bounding schemes can be studied to speed-up the Branch and Bound procedure to solve the MIP.

In this work, we mainly focus on the idea of applying mathematical modeling to decompose time series and a number of improvements are possible. One possible future study on our approach would be considering a fast near-optimal algorithm that can handle fixed-charge constraints (e.g., [20]). In this context, several parameters can be tried using a fast solution method to identify the number of trend shifts and anomalies. Then, this information can be used to solve the proposed MIP to optimality. It would also be interesting to see if the heuristic provides a speed-up in the upper bounding procedure for the MIP.

Bibliography

[1] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, "RobustSTL: A robust seasonal-trend decomposition algorithm for long time series," 12 2018.

[2] C. Hilas, S. Goudos, and J. Sahalos, "Seasonal decomposition and forecasting of telecommunication data: A comparative case study," *Technological Forecasting and Social Change*, vol. 73, pp. 495–509, 06 2006.

[3] B. Arputhamary and D. L. A. Lawrence, "A pragmatic study on time series models for big data," *International Journal of Emerging Research in Management and Technology*, vol. 6, p. 67, 06 2018.

[4] R. Adhikari and R. Agrawal, *An Introductory Study on Time Series Modeling and Forecasting*. 01 2013.

[5] A. Dokumentov and R. J. Hyndman, "Str: A seasonal-trend decomposition procedure based on regression," 06 2015.

[6] S. Alam, Rahman and Parh, "Time series decomposition and seasonal adjustment," *Global Journal of Science Frontier Research: F Mathematics and Decision Sciences*, 2015.

[7] D. G. MacGregor, *Decomposition for Judgmental Forecasting and Estimation*, pp. 107–123. Boston, MA: Springer US, 2001.

[8] R. Fildes, "Regent developments in time series forecasting," *Operations-Research-Spektrum*, vol. 10, pp. 195–212, Dec 1988.

[9] B. Motnikar and T. Pisanski, "Time-series forecasting by pattern imitation — outliers," in *Operations Research '93* (A. Bachem, U. Derigs, M. Jünger, and R. Schrader, eds.), (Heidelberg), pp. 347–350, Physica-Verlag HD, 1994.

[10] M. Theodosiou, "Forecasting monthly and quarterly time series using STL decomposition," *International Journal of Forecasting*, vol. 27, pp. 1178–1195, 10 2011.

[11] A. Korobeinikov, "Financial crisis: An attempt of mathematical modelling," *Applied Mathematics Letters*, vol. 22, no. 12, pp. 1882 – 1886, 2009.

[12] S. Jammalamadaka, J. Qiu, and N. Ning, "Multivariate bayesian structural time series model," *Journal of Machine Learning Research*, vol. 19, 01 2018.

[13] H. Godwin and C. Okafor, "Modified trend and seasonal time series analysis for operations: A case study of soft drink production," *International Journal of Engineering Research in Africa*, vol. 7, pp. 63–72, 09 2012.

[14] M. Sanchez-Vazquez, M. Nielen, G. Gunn, and F. Lewis, "Using seasonal-trend decomposition based on loess (STL) to explore temporal patterns of pneumonic lesions in finishing pigs slaughtered in england, 2005-2011," *Preventive veterinary medicine*, vol. 104, pp. 65–73, 12 2011.

[15] R. B. Cleveland, "Stl : A seasonal-trend decomposition procedure based on loess," 1990.

[16] L. Qin, W. Li, and S. Li, "Effective passenger flow forecasting using STL and ESN based on two improvement strategies," *Neurocomputing*, vol. 356, 05 2019.

[17] W. S. Cleveland, D. M. Dunn, and I. J. Terpenning, "SABL: A resistant seasonal adjustment procedure with graphical methods for interpretation and diagnosis," pp. 201–241, 1979.

[18] H. Bleikh and W. Young, *Time Series Analysis and Adjustment: Measuring, Modelling and Forecasting for Business and Economics*. 01 2014.

[19] F. S. Duarte, R. A. Rios, E. R. Hruschka, and R. F. de Mello, "Decomposing time series into deterministic and stochastic influences: A survey," *Digital Signal Processing*, vol. 95, p. 102582, 2019.

[20] W. E. Walker, "A heuristic adjacent extreme point algorithm for the fixed charge problem," *Management Science*, vol. 22, no. 5, pp. 587–596, 1976.

[21] S. Wamba, S. Akter, A. Edwards, G. Chopin, and D. Gnanzou, "How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study," *International Journal of Production Economics*, 01 2014.

VITA

Şeyma Gözüyılmaz graduated from Milli Piyango Anatolian High School in 2011. She received her B.S. degree in Industrial Engineering from Özyeğin University in June 2016. After graduation, she received her M.S. degree in Business Administration from Uludağ University in June 2018. Besides that, she joined Master of Science program in Industrial Engineering at Özyeğin University, and has been working under supervision of Assoc. Prof. Erhun Kundakcıoğlu. Her research focuses on decomposing time series via mathematical optimization.