Big Data techniques for Solar Power Forecasting

By

Rui Miguel da Cunha Nunes

Dissertation of Master in Data analytics and Decision Support Systems

Supervised by

Ricardo Jorge Gomes de Sousa Bento Bessa
João Manuel Portela da Gama

2017

Biographical Note

Rui Miguel da Cunha Nunes was born in June 30^{th} 1981 in Viseu, Portugal. Following his high school graduation in electronics, he proceeded with his education at Aveiro's University, completing his degree in Applied Mathematics and Computing in July 2005. After graduation, he decided to apply the acquired knowledge in the corporative world. All his experience is related to software developing in all of their stages. These experiences vary in the technologies as well in the business. In 2014 he decided to enrol in the Master in Data Analytics and Decision Support Systems, to proceed with his academic course trying to conjugate both his work experience as well his undergraduate academic course.

Acknowledgements

First and foremost I thank my supervisors for all their help and incentive. Ricardo Bessa for proposed this theme that inspired me from the start. Professor João Gama for all the support that he provided. Special thanks to Laura Cavalcante that shared her knowledge in the LASSO-VAR problem.

I would like to thank my family for all the patient and all the missing hours away from home and some bad humor from the lake of sleep. Thank my boys Miguel e Pedro, for understanding all the missed hours that I was not with you and all joyful moments that I was not there to see.

Abstract

With the growth of social awareness for environmental issues, came a new and important problem to the discussion, related to renewable energies. Although this kind of energy production is already in an advanced state of development there are some issues that still need to be studied and perfected on the real production environment. The Photovoltaic production is gain some importance but the prediction of this kind of energy still need a boost in their performance. The SMART Grid projects brought this issues to another level where the importance of these predictions needs to be as accurate as possible, but they brought another problem, how to accurate this prediction with all data that this kind of problems is associated with. There are some accurate models for a single location prediction but we involving multiple locations in one problem we need to have some computational power. Big Data techniques such the Convex optimization are a field that is important to explore since these techniques applied with the parallel computer programming paradigm can bring that computational power, and more accurate models can be trained faster.
Keywords: Big Data, Convex Optimization, VAR, SMART Grids.

Resumo

Com o crescimento da consciência social para questões ambientais, surgiu um novo e importante problema para a discussão, relacionada às energias renováveis. Embora este tipo de produção de energia já esteja em um estado avançado de desenvolvimento, existem algumas questões que ainda precisam ser estudadas e aperfeiçoadas no ambiente de produção real. A produção fotovoltaica ganha alguma importância, mas a previsão deste tipo de energia ainda precisa de um impulso no desempenho. Os projetos SMART Grid trouxeram esses problemas para outro nível, onde a importância dessas previsões precisa ser o mais preciso possível, mas eles trouxeram outro problema, como precisão dessa previsão com todos os dados associados a esse tipo de problemas. Existem alguns modelos precisos para uma previsão de localização única, mas envolvendo múltiplos locais em um único problema, precisamos ter algum poder computacional. As técnicas Big Data, tais como a Otimização Convexa, são um campo que é importante explorar, uma vez que essas técnicas aplicadas com o paradigma paralelo de programação de computadores podem trazer esse poder computacional e modelos mais precisos podem ser treinados mais rapidamente.
Palavras-chave: Big Data, optimização convexa, VAR, SMART Grids.

Abbreviations

OLS	Ordinary Least Squares
LASSO	Least Absolute Shrinkage and Selection
ML	Machine Learning
NWP	Numerical Weather Prediction
MOS	Model Output Statistics
NN	Neural Networks
ANN	Artificial Neural Networks
AR	Autoregressive
ARMA	Autoregressive and Moving Average
EB	Energy Box
DTC	Distribution Transformer Controller
VAR	Vector Autoregressive
ANN-GA	Artificial Neural Networks trained with genetic algorithms
k-NN	k-nearest neighbors
SVM	Support Vector Machines
ESO	Expected Separable Overapproximation
IEA PVPS	Photovoltaic Power Systems Programme
ECMWF	European Centre for Medium-Range Weather Forecasts
HMM	Hidden Markov model
PV	Photovoltaic
NARX	Nonlinear Autoregressive with eXogenous inputs

Nomenclature

$WN(\sigma)$	**W**hite **N**oise with mean zero and variance σ
ℓ_p	p-Norm equivalent to $\|.\|_p$
$\hat{p}_{t+k\|t,j}$	Prediction on $k-$*ahead* step on site j
doy_k	Day of the year of element k
h_k	Time of the day of element k

Contents

Nota Biográfica i

Acknowledgements ii

Abbreviations ii

Nomenclature iii

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Problem Definition . . . 3
- 1.3 Organization . . . 4

2 State of the art 5
- 2.1 Solar Power predictions . . . 5
  - 2.1.1 Clear Sky models . . . 5
  - 2.1.2 Physical . . . 6
  - 2.1.3 Statistical . . . 7
  - 2.1.4 Hybrid . . . 8
  - 2.1.5 Spatio-temporal . . . 9
- 2.2 VAR . . . 10
- 2.3 ℓ_p Regularization . . . 10
  - 2.3.1 LASSO . . . 11
- 2.4 Convex Optimization . . . 12
- 2.5 Convex Optimization Algorithms . . . 13
- 2.6 Summary . . . 20

3 Contributions to LASSO-VAR Model Fitting 22
- 3.1 Algorithms . . . 23
  - 3.1.1 Shotgun . . . 24
  - 3.1.2 GRock . . . 26
- 3.2 Summary . . . 28

4 Case Study **29**
4.1 Data . . . 29
4.2 Sub-problem . . . 31
4.2.1 RMSE . . . 31
4.2.2 Forecast . . . 36
4.3 VAR . . . 41
4.3.1 RMSE . . . 41
4.3.2 Forecast . . . 48

5 Conclusions and Future work **54**
5.1 Conclusions . . . 54
5.2 Future Work . . . 60
5.2.1 Shotgun . . . 60
5.2.2 GRock . . . 60
5.2.3 General . . . 60

A RMSE Values **61**

B RMSE **67**

C Forecast **75**

Bibliography **81**

List of Figures

1.1 GHI 2
1.2 Problem illustration 3

2.1 LASSO vs. Ridge Regression on $\mathbb{R}^2$ 12
2.2 Coordinate Descent 14

4.1 Monthly Production 30
4.2 Monthly Production 30
4.3 Shotgun (sub-problem) parallel models RMSE for $t+1$ 32
4.4 Shotgun (sub-problem) parallel models RMSE for $t+6$ 33
4.5 GRock (sub-problem) parallel models RMSE for $t+1$ 34
4.6 GRock (sub-problem) parallel models RMSE for $t+6$ 35
4.7 EB17 Shotgun (sub-problem) parallel models'step $t+1$ forecast for 2012-06 36
4.8 EB17 Shotgun (sub-problem) parallel models'step $t+6$ forecast for 2012-06 37
4.9 EB18 Shotgun (sub-problem) parallel models'step $t+1$ forecast for period 2012-06-28/012-07-02 37
4.10 EB18 Shotgun (sub-problem) parallel models'step $t+6$ forecast for period 2012-06-28/012-07-02 38
4.11 EB51 GRock (sub-problem) parallel models'step $t+1$ forecast for 2012-02 39
4.12 EB51 GRock (sub-problem) parallel models'step $t+6$ forecast for 2012-02 39
4.13 EB51 GRock (sub-problem) parallel models'step $t+1$ forecast for period 2012-02-02/012-02-04 40
4.14 EB51 GRock (sub-problem) parallel models'step $t+6$ forecast for period 2012-02-02/012-02-04 40
4.15 Shotgun (VAR) parallel models RMSE for $t+1$ 41
4.16 Shotgun (VAR) sequential models RMSE for $t+1$ 42
4.17 Shotgun (VAR) parallel models RMSE for $t+6$ 43
4.18 Shotgun (VAR) sequential models RMSE for $t+6$ 44
4.19 Problem illustration 45
4.20 Problem illustration 46
4.21 Problem illustration 47
4.22 Problem illustration 48
4.23 EB51 Shotgun (VAR) parallel models'step $t+1$ forecast for 2012-06 . . . 49

4.24 EB51 Shotgun (VAR) parallel models'step $t + 6$ forecast for 2012-06 . . . 50
4.25 EB51 Shotgun (VAR) parallel models'step $t + 1$ forecast for period 2012-06-28/012-07-02 . . . 50
4.26 EB51 Shotgun (VAR) parallel models'step $t + 6$ forecast for period 2012-06-28/012-07-02 . . . 51
4.27 EB41 GRock (VAR) parallel models'step $t + 1$ forecast for 2012-02 . . . 51
4.28 EB41 GRock (VAR) parallel models'step $t + 6$ forecast for 2012-02 . . . 52
4.29 EB51 GRock (sub-problem) parallel models'step $t + 1$ forecast for period 2012-02-02/012-02-04 . . . 52
4.30 EB51 GRock (sub-problem) parallel models'step $t + 6$ forecast for period 2012-02-02/012-02-04 . . . 53

5.1 Shotgun (Sub-problems) parallel RMSE for step $t + 6$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 10$ . . . 56
5.2 Shotgun (sub-problem) sequential RMSE for step $t+6$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$ . . . 57
5.3 Shotgun (VAR) parallel RMSE for step $t+6$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$ 58
5.4 Shotgun (VAR) sequential RMSE for step $t + 6$. Models $\lambda = 0, \lambda = 10$ *and* $\lambda = 25$ . . . 59

B.1 Shotgun (sub-problem) sequential models RMSE for $t + 1$ . . . 67
B.2 Shotgun (sub-problem) Sequential models RMSE for $t + 6$ . . . 68
B.3 GRock (sub-problem) parallel models RMSE for $t + 1$ . . . 69
B.4 GRock (sub-problem) parallel models RMSE for $t + 6$ . . . 70
B.5 Shotgun (sub-problem) sequential RMSE for step $t+1$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$ . . . 71
B.6 Shotgun (VAR) sequential RMSE for step $t + 1$. Models $\lambda = 0, \lambda = 10$ *and* $\lambda = 25$ . . . 72
B.7 Shotgun (Sub-problems) parallel RMSE for step $t + 1$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$ . . . 73
B.8 Shotgun (VAR) parallel RMSE for step $t+1$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$ 74

C.1 EB17 Shotgun (sub-problem) parallel models'step $t+1$ forecast for 2012-06 75
C.2 EB18 GRock (sub-problem) Sequential models'step $t + 1$ forecast for period 2012-02-02/2012-02-04 . . . 76
C.3 EB18 GRock (sub-problem) Sequential models'step $t + 6$ forecast for period 2012-02-02/2012-02-04 . . . 76
C.4 EB18 GRock (VAR) Parallel models'step $t + 1$ forecast for 2012-09 . . . 77
C.5 EB18 GRock (VAR) Parallel models'step $t + 6$ forecast for 2012-09 . . . 77
C.6 EB18 GRock (VAR) Sequential models'step $t + 1$ forecast for 2012-09 . . . 78
C.7 EB18 GRock (VAR) Sequential models'step $t + 6$ forecast for 2012-09 . . . 78
C.8 EB1 Shotgun (VAR) Parallel models'step $t + 1$ forecast for 2013-01 . . . 79

C.9 EB1 Shotgun (VAR) Parallel models'step $t + 6$ forecast for 2013-01 . . . 79
C.10 EB1 Shotgun (VAR) Sequential models'step $t + 1$ forecast for 2013-01 . . 80
C.11 EB1 Shotgun (VAR) Sequential models'step $t + 6$ forecast for 2013-01 . . 80

List of Tables

2.1 Algorithms in Literature 21

5.1 GRock Models'RMSE for step $t + 1$ 55
5.2 Shotgun Models'RMSE for step $t + 1$ 55

A.1 RMSE for Shotgun (sub-problem) parallel models for step $t + 1$ 61
A.2 RMSE for Shotgun (sub-problem) parallel models for step $t + 1$- Cont. . . 62
A.3 RMSE for Shotgun (sub-problem) sequential models for step $t + 1$ 63
A.4 RMSE for Shotgun (sub-problem) sequential models for step $t + 1$ - Cont 64
A.5 RMSE for Shotgun (VAR) parallel models for step $t + 1$ 64
A.6 RMSE for Shotgun (VAR) parallel models for step $t + 1$ - cont 65
A.7 RMSE for Shotgun (VAR) sequential models for step $t + 1$ 65
A.8 RMSE for Shotgun (VAR) sequential models for step $t + 1$ - Cont 66

Chapter 1

Introduction

With the growing environmental concern, renewable energy sources had gain importance and left out the marginal presence in the energy production. Wind energy has the higher usage, but solar power had gained a new importance and even there are some benefits by governmental promotion. Under DL 363/2007, consumers could become micro-producers by selling part of the energy produced by his photovoltaic system. In another scale of production, as the price of photovoltaic system prices decreases the number and power production installation increases according to with IEA PVPS at the end of 2014 at least 38.7 *GW* have been installed bring the total power installed all over the world to 177 *GW*. They claim also that 19 countries have now photovoltaic installations to cover 1% of their annual electricity demand, Portugal is among this countries. Technology that allows an affordable solar power production has been under research, Singh (2013) present a review of this research, in that work the author state that the solar energy role in global energy production is increasing.

In Singh (2013) claims that photovoltaic manufacturing costs did not reach yet the point where this technology can replace the conventional production, dependent on fossil fuels. A particular cost is the storage of this kind of energy that is very expensive. This type of installations benefits from its inclusion on a Grid, like a smart grid. A smart grid is an improvement of the *Classical* electrical grid, this grid provides a better understanding of production and demand of electricity. In case of solar power, this grid brings the advantage of distribution if this kind of energy with higher effectiveness.

Solar power production is slightly different from the traditional forms (and non-renewable) of energy productions. In this form of energy production, we have to take in account it's *variability*, along the day from sun rise to sun set, the amount of energy produced is different, and the appearance of clouds affects also the production. Besides *variable* this kind of production is also *uncertain*, by uncertain, we mean that the forecast in advanced steps in this production is not perfect.

As we can see in figure 1.1, south Europe has a good solar exposure given the ability to solar power production. Portugal has an overall good exposure, especially in the south. It will interest to study the possibility of taking advantage of this natural resource in order to

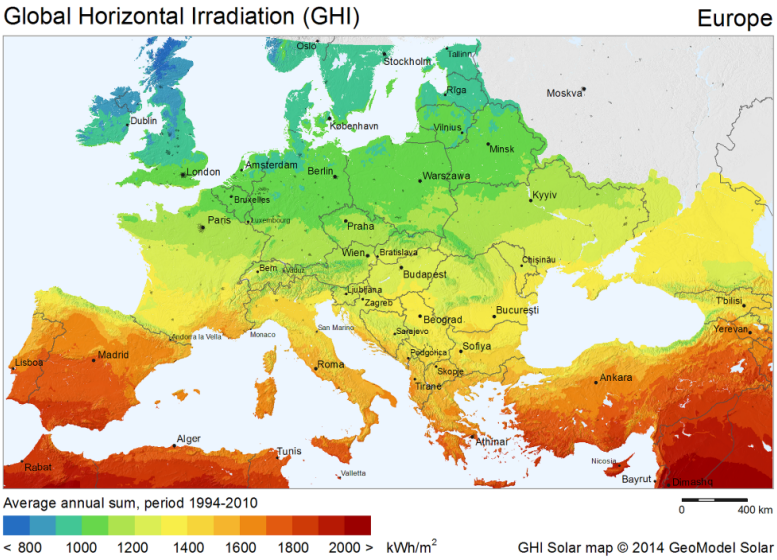


Figure 1.1: Global Horizontal Irradiation

relieve our energy dependence and a long term project maybe could step to be a provider of electricity to Europe grid.

In terms of the predictions of this kind of energy production, there are some interesting studies and approaches to lead to better models. These studies go from the usage of ML algorithms such in Ercan İzgi (2012), where the authors Artificial Neural Networks (ANNs) into the predictions at a small scale. In cloud imagery, Chow et al. (2011) studied the usage of ground-based total sky imager to forecast cloud. Golestaneha et al. (2015) studied the PV generation uncertainty and its dependency to time and space.

This approaches and several others will be addressed in the following chapters. The main idea that it is interesting is that the major works on PV energy forecast are based in a single location, but with the growth of the SMART Grids there was the need to address this problem with another angle where we look into all locations of that SMART Grid and make predictions for all using Data from all locations such as in Bessa et al. (2015)

1.1 Motivation

The growth of Solar production brings new concern and importance to forecast of this production. Since the storage of this kind of energy is expensive, a smart use of this energy must be applied, the accuracy and reliability of forecast has a huge importance in order to provide a better plan both in distribution of this energy, and in case of domestic use, a better plan in house consumption.

This forecast beyond his accuracy and reliability has to be fast, distributors can't wait a large time period in order to plan his next steps. When in a Grid forecasts can take advan-

tage of all grid information to improve it's accuracy, in photovoltaic power all historical information of all production sites can be used with better results than when analysing a single site. But when using more information we get a new problem, in this case a computational problem, were the execution time will grow as the information to study grow.

Convex optimization techniques applied to this data will bring a new approach in this problems, with improvement of execution time, when this techniques are applied in a parallel computing, this way we are able to perform more tasks simultaneously.

1.2 Problem Definition

In this work, we will deal with solar power predictions in a ***very-short term*** . Since a amount of information necessary in several cases is huge, we will addressed this with Big Data techniques special focus on Convex optimization to solve this kind of problems.

We are interested in producing forecasts for the next 6 steps ahead, considering the *VAR* framework. In order to estimate the coefficients of that framework we will considering *LASSO* estimator. To solve this minimization problem we will take advantage of its convexity and apply some coordinate descendent method in a parallel computing.

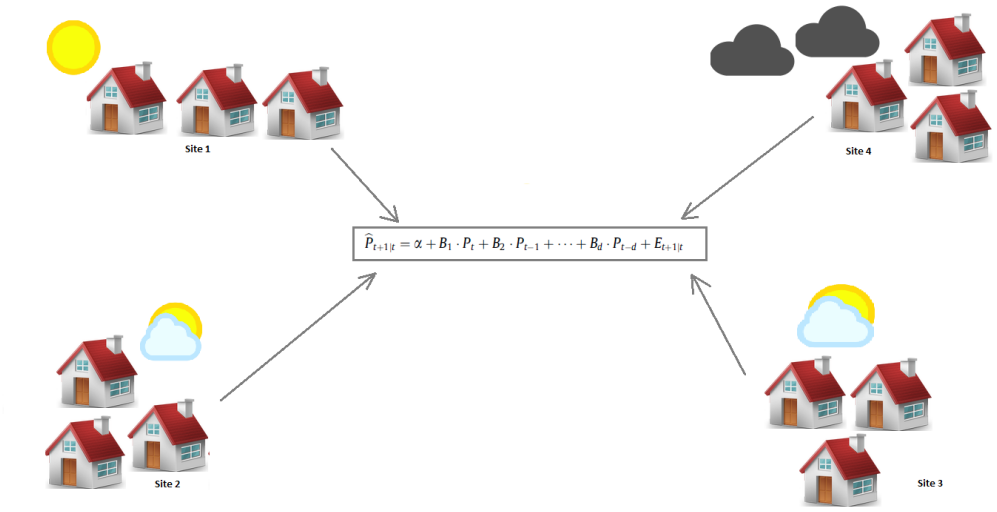


Figure 1.2: Problem illustration

In figure 1.2 are represented 4 sites with different sunny conditions. What we intend to perform is the forecast solar power in each one of this sites considering everyone of this sites, $\hat{p}_{t+k|t,j}$. By doing so we can inflict and predict the effect of clouds of *site 4* in the remaining sites, and the effect of the clouds in *sites 2* and *3*, somehow forecast the overall effect of all weather conditions in to all sites.

1.3 Organization

This dissertation is structured in 3 major chapters. In chapter 2 is described a literature review in the several knowledge areas that this work involves. In the first section of this chapter it is presented the literature review on the Solar Power predictions in the multiple knowledge, such as Spatio-Temporal, statistical, etc. On the second section of that chapter is presented the VAR problem. In the third section is presented the ℓ_p regularization focusing in the LASSO problem, this problem is a ℓ_1 regularization. In the fourth and fifth sections are presented the general Convex Optimization problem and the some algorithms that can be applied in those kind of problems.

After this literature review it is presented the contributions that this work present, these contributions are described on chapter 3. These contributions were made on the LASSO-VAR problem and the adaptation on the algorithms to this specific problem. In this work the focus was on the Shotgun an GRock algorithms.

In chapter 4 are presented the case study were those contributions were applied. This chapter is split in 3 sections. In the first section is described the used that to train and test the models. Second and third sections described some of the results of those models. These sections are unfolded into the analysis of the RMSE and the forecast obtained by the models.

In the last chapter are described some conclusion about the models and problem approaches of this work. In the end of that chapter will be presented some future work to advance with the study of the problem.

Since the problem described on section 1.2 involves a huge amount of Data this work aims to contribute in the Convex optimization algorithms that can be applied into that problem. The problem can be defined into a LASSO-VAR problem. The algorithms studied were focused in the parallelization property of those algorithms. This property brings some improvements into the obtained models and in same cases the runtime on the model training.

Chapter 2

State of the art

Due to the nature of this work, it will be presented a state of the art in two fields of research. We saw on the problem definition how they connect. Basically we will see some methods applied to ***Solar Power forecasting*** and ***Convex Optimization***.

Considering the problem that will be studied, it will be given a major attention to Convex optimization coordinated methods.

2.1 Solar Power predictions

Monteiro et al. (2009) presented a split solar power forecasting in three different time horizons: **very short-term** (*up to 6 hours ahead*), **short-term** (*up to 3 days ahead*) and **Medium term** (*up to 7 days ahead*). Each one of this horizons has it's own set of techniques and methods to answer theirs specificity. The Medium term due to his increase on time it's error rate also increase.

Besides time horizons solar forecast can also be categorized by its nature. Diagne et al. (2012) presented 3 different categories (i) Physical, (ii) Statistical and (iii) Hybrid.

For *short-term* there are several studies with a different approach. Monteiro et al. (2009) presented a study that intended to evaluate combined forecasting and proposed three different approaches to the problem, (a) *top-down* , (b) *bottom-up* and (c) *regression*, methods. The most recent studies try to combine Statistical/Machine Learning with NWP. A two step approach was proposed by Bacher et al. (2009). The method presented start by applying the *clear sky* model, this step will remove the diurnal component of the solar generation. Then apply an autoregressive model with exogenous input (ARX) to past observations with NWP.

2.1.1 Clear Sky models

Clear Sky models are based on the cloudless situation, there are physical properties of the sky also beside the clouds that affect the clearness of the sky but there aren't visible to our

eyes. These models are applied in several methods either physical or statistical. Inman et al. (2013) present several methods that can be applied in different models. Clear Sky models are applied in order to obtain an estimate of a clearness index. A clear-sky model must be properly calibrated to provide an accurate measure of the clearness index. In their foundations, these models are the "normalizers" of the solar information, from satellite to descriptive information as weather or historical information.

In solar forecasting there are two indexes that we can use, clear sky index k_t and clearness index K_t they are similar, the difference between them is the ratio it self. Clear sky index measures the ratio between the radiation with the clear sky radiation at the ground. And the clearness index measures the ratio between the measured ratio with the extraterrestrial irradiance.

The clear sky model that we will take into account and implement in our work, is a based on weighted quantile regression presented in Bacher et al. (2009) and is defined by

$$\hat{p}_t^{cs} = \arg\min_{\hat{p}_t^{cs}} \sum_{i=1}^{N} K(h_t, doy_t, h_i, doy_i) \cdot \rho(\tau, e_i) \tag{2.1}$$

were $K(h_t, doy_t, h_i, doy_i)$ is the kernel product of the predictors h and doy, that weights each observation and $\rho(\tau, e_i)$ is a loss function of the quantile regression problem. Remark that $e_i = p_t - \hat{p}_t^{cs}$ and

$$\rho(\tau, e_i) = \begin{cases} \tau \cdot e_i & , e_i \geq 0 \\ (1-\tau) \cdot e_i & , e_i < 0 \end{cases} \tag{2.2}$$

Being circular variables, then a circular kernel is used:

$$K(x_t, x_i, \sigma) = e^{\frac{1}{\sigma} \cdot \cos\left[2\pi \cdot \frac{(x_t - x_i)}{d}\right]} \tag{2.3}$$

were σ is smoothing parameter and d is variable x periodicity. In equation (2.1) $K(h_t, doy_t, h_i, doy_i)$ is given by:

$$K(h_t, doy_t, h_i, doy_i) = \frac{K(h_t, h_i, \sigma_h) \cdot K(doy_t, doy_i, \sigma_{doy})}{\sum_{i=1}^{N} \left[K(h_t, h_i, \sigma_h) \cdot K(doy_t, doy_i, \sigma_{doy}\right]} \tag{2.4}$$

2.1.2 Physical

Physical model uses weather information and cloud images. Several studies have been made in this fields, cloud motion in satellite imagery performs better than NWP (Numerical Weather Predictions) in a shorter time horizon 3-4 hours because even NWP is capable of forecasting clouds several days ahead the time arrival of that clouds is only accurate in short term within several hours. Cloud imagery can be obtained through Satellite or ground-based sky camera. Satellite imagery forecast is a challenge due to its coarse spatial resolution.

Cloud imagery forecast, these images are obtained through satellite or ground-based, are based on Motion vectors to detect and track the motion of the clouds in images. Chow et al. (2011) shown that the use of ground-based total sky imager in forecast cloud up to 10 minutes ahead with some limitations, there is the fact that capturing deterministically, at a fine spatial scale, low clouds and large clouds variability is near impossible with NWP and satellites. Besides this, there was also a problem with shadows and obscuration. Jayadevan et al. (2012) produced a similar study recurring to a sun-tracking camera, with higher resolution, compared with an all-sky image, near the sun and the resolution is independent of time of the day and season. This described an intermittency forecast method up to 10 minutes, this study showed that under ideal situation it can provide accurate forecast 10 minutes ahead within 1 minute.

In physical models, we can also find NWP models, that uses weather information to build the model. The basis of this models are the forecast of the atmosphere state, through a set of differential equations that we can see in Lorenz and Heinemann (2012), that describe the physical laws of weather. This kind of models can be categorized in to *Global* or *Mesoscale* Models. Diagne et al. (2013) refer that Global NWP are in operation at 15 weather services. Some examples of global NWP are GFS from USA and ECMWF (European Centre for Medium-Range Weather Forecasts). Even with the increasing in the resolution in the last few years, global NWP has a low resolution and it's impossible to obtain a detailed mapping for a small-scale. In order to perform forecast on this scale, we might apply a Mesoscale model also referred in the literature as a regional model. As the name of the model indicates, this kind of models covers a region, a part of the Earth, even though allow a higher spatial resolution. NWP output can be refined through Postprocessing methods, this method could be applied to reduce forecast errors, introduce local effects, etc. There are several approaches in Postprocessing, the mainstream approach is *MOS*. Diagne et al. (2013) refer also *Kalman filter*, *Temporal interpolation*, *Spatial averaging* and *Physical post processing approaches*.

2.1.3 Statistical

This type of models are based on historical data, and we can classify as statistical and learning methods. In learning methods, we find ML algorithms, genetic algorithms, NN, etc. The major studies in this kind of models are focus on the analysis of univariate Time Series and more recently ML algorithms. They focus only on the historical information of each Site.

From a statistical point of view, the classical algorithms such AR, ARMA, Markov Chains, etc. These methods have been shown as good predictive models in terms of time series predictions. A $AR(p)$ model, predict moment t given the last p moments is given by (2.5), were b_i are the coefficients and $\epsilon_t \sim WN(\sigma_t^2)$

$$y_t = c + b_1 y_{t-1} + \cdots + b_p y_{t-p} + \epsilon_t \tag{2.5}$$

ARMA is a combination of two models ($AR(p)$ and $MA(q)$), $ARMA(p, q)$, is a model with p autoregressive terms and q moving-average terms, are given by (2.6), were b_i are the AR coefficients, θ_j are the MA coefficients and $\epsilon_k \sim WN(\sigma_k^2)$

$$y_t = c + \epsilon_t + \sum_{i=1}^{p} b_i y_{t-i} + \sum_{j=1}^{q} \theta_i \epsilon_{t-j} \tag{2.6}$$

Since this kind of models requires a stationary time series. One way to test the stationary of a Time Series is applying an ADF (Augmented Dickey-Fuller) test. Diagne et al. (2012) state that this test applied to a solar time series shown a non-stationary times series. Due this result there's a preprocessing that is needed perform. There are the need to normalize the time series. The most common method in literature is applying a *Clear Sky* model. The output from the clear sky model, y_t^{cls}, is applied to normalize the obtained measure of solar power y_t:

$$y_t^{norm} = \frac{y_t}{y_t^{cls}} \tag{2.7}$$

Besides the classical models, more recent studies are applying ML algorithms and techniques in order to predict solar power. Many ML algorithms have been study in solar power forecasting, Mellit and Pavan (2010) presented a study applying *ANN* in a $24h$ forecasting. They showed that this technique has advantages over some other techniques, but this forecasting as a downside that is the computing time since it takes about 10 minutes to provide a forecast. They intend to study the implementation a **ANN-GA** in order to decrease this time.

Huang and Perry (2015) studied the application of *k-NN*, this study already has in consideration the time constraint on the forecast and have been implemented in a parallel way. They used Gradient boosting to convert NWP output to point forecasts of solar power and Fourier transformation to de-trending data. Then in order to make predictions, they applied k-NN regression. These methods produced an accurate forecast of solar power.

Zeng and Qiao (2013) shown that *SVM*, in terms of accuracy, provides better forecast than *AR* since these methods capture the nonlinearity and time variations of solar radiations.

2.1.4 Hybrid

This type of models was presented to improve the accuracy of the forecast and overcome some issues that individual model could present. There are several studies in these models with a higher accuracy than the individual forecasts.

Cao and Cao (2005) and Cao and Cao (2006) presented studies were a combination between ANN and wavelet analysis. ANN is by itself a powerful tool to forecast solar irradiance, this studies shown that the combination with wavelet analysis improves the accuracy of such forecasts. The wavelet analysis is used in a preprocessing data to then applied to ANN in order to perform the forecast. According to with Cao and Lin (2008)

although ANN, in terms of accuracy, performs better than traditional prediction models, ANN-based models are yet reasonable in a forecast precision that researchers and engineers are looking for. So in order to address this, they presented a new study to improve this forecasts. They study the appliance of a dynamic characteristic of a recurrent neural network (*RNN*) instead of an ANN, with the ability to catch nonlinearity that wavelet neural network(*WNN*). They even present a new concept Diagonal recurrent wavelet neural network (*DRWNN*) to deal with the fine forecast.

In the same line of study (combining ML algorithms with statistical methods) Ji and Chee (2011) presented a Study were its combined *ARMA* model with time delay neural network (*TDNN*). By themselves *ARMA* a *TDNN* are very powerful, simulations performed in their study show that this hybrid model gives "excellent" results.

In another filing of study Marquez et al. (2013) presented a study that combines a physical model with a stochastic learning. In this work, the authors use satellite images analysis as input to ANN model. This analysis included velocimetry and cloud index. According to with the authors, this was the first attempt to join stochastic learning with image processing in order to solar irradiation forecast.

2.1.5 Spatio-temporal

Although Lorenz et al. (2009) presented a study in physical approach provide by *ECMWF*, they show that forecast accuracy for an ensemble of a distributed system is higher than the forecast for a single system. This implies that the knowledge of site environment brings a better forecast performance.

The spatial-temporal models are a "recent" field of study, that take advantage of site environment and from their historical information. This field of study is from statistical models. In this field of study instead of analysing only *one* site, we are looking to its neighbours also. Lonij et al. (2012) produced a study where they compare a method using a Network of residential PV system with an NWP and ground-based camera methods. They have shown that by using a network of PV systems, the physical forecasting methods were outperformed for a 15-min interval.

Meteorological phenomena have a high geographical auto-correlation, and so the way that PV systems are formed (centralized or distributed) have an affect over the *variability* and *uncertainty*. Tabone and Callaway (2015), presented a study where they applied a *HMM*(hidden Markov model) to predict this characteristics. They intended to deal with the reserve requirements. They discover that under certain conditions the variability distribution approaches to a Gaussian distribution. They presented the model as a planning tool for additional reserve capacity and\or to balance in a spatial area (large and small) the variability characteristic.

Golestaneha et al. (2015) studied the strong dependency of PV generation uncertainty to time and space. To address this situation they had analysed and captured spatio-temporal dependencies in PV generation. They claim that in an operational environment that is more sensitive in structure dependence, when space-time correlation is unknown,

using probabilistic forecast may provide a suboptimal solution. They verified also that even when historical data is unavailable in order to find structure dependency of PV generations, an alternative approach might be modelling covariance matrix recursively.

Bessa et al. (2015) applied a *VAR* framework in order to include the neighbours information in the forecast. This work applied two different fit methods to VAR framework and used the clear sky model presented by Bacher et al. (2009), based on weighted quantile regression. They showed that NARX (Non-linear Autoregressive with eXogenous inputs) model outperform persistent methods in 15-min and beyond forecasts. They confirmed that using information of neighbouring improves the forecast accuracy.

Vaz et al. (2016) presented a model for PV prediction with a *NN* architecture, using a NARX model, using not only the site meteorological data but also their neighbouring data. The NARX model presented to be also verify effective in multi-step predictions.

2.2 VAR

The *vector autoregression* (VAR) model were introduced by Sims (1980) and are a extension of *Autoregressive* (AR) models, to model multivariate time series. The VAR model has some descriptive power as well a good forecasting. It often provides a better forecast result than the AR model. The flexibility of VAR happens because of the conditionality on potential future paths of the same variable in the model.

An n-variable vector autoregression of order p VAR(p), is a system of n linear equations, on witch every equation describes the dynamics between one variable as a linear function of the p previous lags of n variables of the system. A $p-th$ order *VAR* is:

$$y_t = c + B_1 y_{t-1} + B_2 y_{t-2} + \cdots + B_p y_{t-p} + e_t \tag{2.8}$$

or in a matrix notation we have:

$$Y = BZ + U \tag{2.9}$$

We can estimate $\hat{B}$ by applying a classical approach with the OLS. We can also estimate using the LASSO approach that will lead to a sparser solution and in the same situation a better result to this estimation.

2.3 ℓ_p Regularization

In Machine Learning and statistics there are some regularization methods, in order to prevent over-fitting of the model to data. In machine learning the most common regularizations are ℓ_1 and ℓ_2. Then instead the ML algorithm minimize the lost function $f(X)$, this is regularized by adding is a penalty. This means that the problem is now the minimization of new penalized function with ℓ_p regularization, $f(X) + \lambda \|.\|_p$ where $\lambda \geq 0$.

ℓ_1 regularization are often preferred methods, his main advantage is a sparse model that is produced thus performs feature selection within the learning algorithm. There is a drawback in this regularization, ℓ_1 is not differentiable, and this might require changes to the learning algorithms.

2.3.1 LASSO

LASSO is an ℓ_1 regularization to a linear regression. This method was presented by Tibshirani (1994).

In a usual regression situation, where data is in the form : $\left(X^i, y_i\right), i = 1, ..., N$ where $X^i = \left(x_{i1}, ..., x_{ip}\right)^T$ are the predictor variables and y_i are the responses OLS estimates by minimizing the residual squared error. There are some techniques to improve this estimate, the standard was *Subset Selection* and *Ridge Regression*, but both of this techniques has some positive aspects and some issues. *Subset Selection* give a very interpretable model but is a very variable technique due to its descriptive nature the feature is retained or dropped from the model, so with small changes in the data we might select very different models and so reduce its prediction accuracy. The *Ridge Regression* is a more stable technique because is a continuous process that shrinks coefficients, but this does not set any coefficient to 0 and the models are not so easily interpretable.

Tibshirani (1994) presented a new technique LASSO (***L**east **A**bsolute **S**hrinkage and **S**election **O**perator*), this technique combines the best features from *Subset Selection* and *Ridge Regression*.

Let's assume that the observations are independent or y_i's are conditionally independent given x_{ij}'s. Let's assume also that x_{ij} are standardized so that $\sum_i \frac{x_{ij}}{N} = 0$ and $\sum_i \frac{x_{ij}^2}{N} = 1$. Then LASSO estimate is given by equation (2.10)

$$\begin{aligned} \left(\hat{\alpha}, \hat{\beta}\right) = \quad & \arg\min \quad \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \\ & \text{s.t.} \qquad \|\beta\|_1 \leq t \end{aligned} \tag{2.10}$$

were $\hat{\beta} = (\hat{\beta_1}, ..., \hat{\beta_p})^T$ is the LASSO estimate. To control the shrinkage that is applied to the estimates its used the parameter $t \geq 0$. For all t, the solution for α is $\hat{\alpha} = \bar{y}$. Without losing generality we can assume that $\bar{y} = 0$ and hence omit α, so $\hat{\beta}$ is estimate by:

$$\begin{aligned} \hat{\beta} = \quad & \arg\min \quad \sum_{i=1}^{N} \left(y_i - \sum_j \beta_j x_{ij} \right)^2 \\ & \text{s.t.} \qquad \|\beta\|_1 \leq t \end{aligned} \tag{2.11}$$

The criterion $\sum_{i=1}^{N} \left(y_i - \sum_j \beta_j x_{ij} \right)^2$ is equal to the following quadratic function:

$$\left(\beta - \hat{\beta}^0\right)^T X^T X \left(\beta - \hat{\beta}^0\right) \tag{2.12}$$

In figure 2.1 it's represented the elliptical contours of equation (2.12) in $\mathbb{R}^2$, centred in the OLS estimate $\hat{\beta}$.

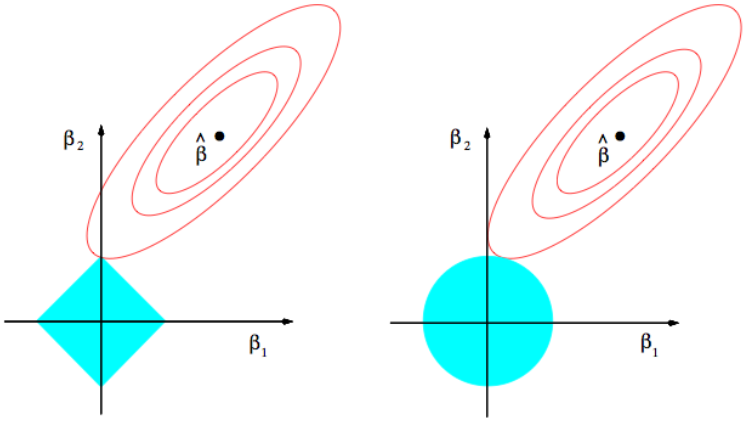


Figure 2.1: (*Left*) LASSO; (*right*) *Ridge Regression*

The problem solution is the first place where the contours intersect restriction region. As we can see in figure 2.1 LASSO constrain region (given by $\|\beta\|_1 \leq t$) is a *rotated square*, then sometimes this solution occurs at a corner corresponding to a zero coefficient. On the other hand, if we consider *Ridge Regression* restriction region (given by $\|\beta\|_2^2 \leq t$) has no corners hence this will rarely produce solutions with zero coefficients.

2.4 Convex Optimization

In convex Optimization there are different method such as Interior Point, Newton, ect., for a more profound knowledge the book from Boyd and Vandenberghe (2004) present a good base in *Convex Optimization*. In this work we will be focus on *Coordinate Descendent* methods, since this methods are more capable to preform on a parallel computing. In this chapter lets consider the following problem:

$$\min_{x \in \mathbb{R}^n} F(x) = \quad f(x) + \lambda \cdot \mathcal{R}(x) \tag{2.13}$$

Cevher et al. (2014) presented the 3 major pillars for Convex Optimization in Big Data. This pillars are:

- **First-order methods:** this is theoretically robust to approximation. They produce numerical solutions with low/medium accuracy, they are nearly dimension-independent in theirs convergence rates.
- **Randomization:** Since we can control their expected behaviour this approximation technique enhance the scalability of the first-order methods

- **Parallel and distributed computation:** First-order methods have a flexible framework that allows distributing optimization task through parallel computations.

Let's analyse each one of this pillars. Considering a special case of equation (2.13), where the objective function is a differentiable convex function$f\ F(x) = f(x)$, the first order method applied to this case is the *gradient method*. This method iteratively updates x as:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) \tag{2.14}$$

where k is the counter and α_k is the step-size that ensures convergence.

When we consider the Composite objective like equation (2.13), where F is composed by a differentiable convex function f and a non-smooth convex function $\mathcal{R}$, the non-smooth part of the problem could reduce efficiency. The *Proximal-gradient* methods take advantage of the composite structure of the problem they retain the convergence rate of the gradient method for the *smooth* part of the problem. It appears that this is a natural extension of the gradient methods when equation (2.14) is seen as an optimization problem:

$$x^{k+1} = arg \min_{y \in \mathbb{R}^n} \left\{ f(x^k) + \nabla f(x^k)^T \left(y - x^k\right) \frac{1}{2\alpha_k} \left\|y - x^k\right\|_2^2 \right\} \tag{2.15}$$

Considering now the problem as an all, we can define the proximal by simply include the *non-smooth* part of the problem:

$$x^{k+1} = arg \min_{y \in \mathbb{R}^n} \left\{ f(x^k) + \nabla f(x^k)^T \left(y - x^k\right) \frac{1}{2\alpha_k} \left\|y - x^k\right\|_2^2 + g(y) \right\} \tag{2.16}$$

So the update rule of the proximal-gradient method is given by equation (2.16) or in a reduced form:

$$x^{k+1} = prox_{\alpha_k g}\left(x^k - \alpha_k \nabla f(x^k)\right) \tag{2.17}$$

In theory, the first-order methods are good methods for very large-scale problems, but in practice, the exact numerical computations iterations demanded by them can make these simple methods infeasible, as the dimensions of problems grow. Hence these methods as very robust using approximations of their optimization primitives Schmidt et al. (2011),

2.5 Convex Optimization Algorithms

Coordinate Descent method, are based on the concept that the minimization of a multivariate function $G(X)$ can be achieved by solving univariate (or a simplest problem) optimization problem in a loop. Figure 2.2 represent this idea.

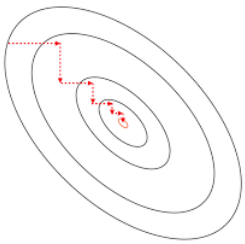


Figure 2.2: Coordinate Descent

In this section, we will study some algorithms that try to deal with the problem given by (2.13) in a parallel manner. In the end of the chapter will be presented a summary of the algorithms in table 2.1. For each one of this algorithms, we will consider the base problem given by (2.13) and present some restrictions and conditions to the algorithm. The base algorithm is RCDM, that present the first study of random select the coordinate to update.

Random Coordinate Descent Method (RCDM)

In Nesterov (2010) was presented **RCDM**, for problems where the regularization function is zero, $\mathcal{R}(x) = 0$. He define the optimal coordinate step as

$$T_i(x) \overset{def.}{=} x - \frac{1}{L_i} U_i \nabla_i f(x)^{\#} \tag{2.18}$$

It was also define a random counter $\mathtt{R}_\alpha$, that generate integer value $i \in 1, \cdots, n$ with a probability:

$$p_\alpha^{(i)} = L_i^\alpha \left[\sum_{j=1}^{n} L_j^\alpha \right]^{-1} \tag{2.19}$$

So any operation $k = \mathtt{R}_\alpha$ means that k is an integer value from $1, \cdots, n$ is chosen with a probability define in equation (2.19). A Special case is when $\alpha = 0$ then $\mathtt{R}_0$ generates a uniform distribution. Knowing step-size and the probability of choice we can define the method:

Algorithm 1 Random Coordinate Descent Method

1: choose $x_0 \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$
2: $for\ k \geq 0$
3: $choose\ i_k = \mathcal{R}_\alpha$
4: $update\ x_{k+1} = T_{i_k}(x_k)$

Shotgun

In Bradley et al. (2011) was presented the algorithm **Shotgun**, this algorithm was based on Stochastic Coordinate Descent (**SCD**) from Shalev-Shwartz and Tewari (2009). This study was for ℓ_1 regularization function, $\mathcal{R}(x) = \|x\|_1$. The main idea in this algorithm is applying the coordinate update not only for one but P coordinates uniformly selected, where P is the number of processors in the machine. The algorithm 2 we can see each one of the steps where δ_{x_j} indicates the update that is perform in coordinate j

Algorithm 2 Shotgun

1: choose $P \geq 1$, the number of parallel updates
2: choose $x_0 \in \mathbb{R}^{2d}_+$
3: **while** not converge **do** (*In parallel on P processors*)
4: choose $j \in \{1, \cdots, 2d\}$ uniformly at random
5: set $\delta_{x_j} \leftarrow max\left\{-x_j, \frac{-(\nabla F(x))_j}{\beta}\right\}$
6: update $x_j \leftarrow x_j + \delta_{x_j}$
7: **end while**

This algorithm was applied to LASSO and its performance was measure against others LASSO solvers. Shotgun performs well and the converging rate is faster then others, particularly in Large sparse datasets were most algorithms fail.

GRock

Let's consider that $\mathcal{R}(x)$ is separable and $\mathcal{R}(x) = \sum_{i=1}^{n} r(x_i)$. Let's consider also that $f(X) = \mathcal{L}(\mathbf{A}x, b)$ and for simplicity $\mathcal{L}$ is convex and A has columns with unit ℓ_2-norm. Let $\beta > 0$ such that:

$$\mathcal{L}(\mathbf{A}(x + d), b) \leq \mathcal{L}(\mathbf{A}x, b) + g^T d + \frac{\beta}{2} d^2$$

where $g = \mathbf{A}^T \nabla \mathcal{L}(\mathbf{A}x, b)$, qe can define to each coordinate i it's potential:

$$d_i = arg \min_d \lambda \cdot r(x_i + d) + g_i d + \frac{\beta}{2} d^2 \tag{2.20}$$

where g_i is the i^{th} entry of g. We can express equation (2.20) in it's closed form $d_i = \mathbf{prox}_{\frac{\lambda}{\beta} r}\left(x_i - \frac{1}{\beta} g_i\right) - x_i$. This algorithm considers the division of coordinates into N blocks, and for each j block:

$$m_j = max\left\{|d| \ : \ d \text{ is an element of } d_j\right\} \tag{2.21}$$

GRock updates the best coordinate of the best P blocks. Let s_j such that $m_j = d_{s_j}$, this means that s_j achieves it's maximum.

This algorithm is not parallel in terms of computing, but the paper gives the way of parallelization of the algorithm steps 3 and 4 can be made in parallel and then step 5 could be executed through MPI and finally step 6 can be preformed in parallel also.

Algorithm 3 GRock

Initialize $x = 0 \in \mathbb{R}^n$
while not converge **do**
    $d_j \leftarrow$ (2.20), for each block j
    $m_j, s_j \leftarrow$ (2.21), for each block j
    $\mathcal{P} \leftarrow$ the indices of P blocks with largest m_j
    $x_{s_j} \leftarrow x_{s_j} + d_{s_j}$, for each $j \in \mathcal{P}$
end while

Nonuniform SYNchronous Coordinate descent

Richtárik and Takác (2013) presented 'NSync, this algorithm as restriction $f(x)$ must be strongly convex and smooth. This algorithm unlike the others that we saw select the coordinates not uniform way.

The first step in this algorithm is to set a probability p_S to each subset S of $[n] := \{1, \cdots, n\}$ under the condition $\sum_S p_S = 1$. It's also selected a stepsize $w_i > 0,\ i = 1, \cdots, n$ and in each iteration there is generated a random sample $\hat{S}$ independent from previous iterations following the law $\mathbf{Prob}(\hat{S} = S) = p_s$ and then the coordinates $i \in \hat{S}$ are updated in parallel. The sampling of $\hat{S}$ is *non-uniform*, so $p_i := \mathbf{Prob}(i \in \hat{S}) = \sum_{S: i \in S} p_s$ can vary with i. We assume that $e^i \in \mathbb{R}^n$ is the i-th unit coordinate vector. The algorithm is in the form:

Algorithm 4 'NSync

Set $x^0 \in \mathbb{R}^n$, a subset $\{p_s\}$, and the stepsizes $w_1, \cdots, w_n > 0$
for $k = 0, 1, \cdots$ **do**
    select random set coordinates $\hat{S} \subseteq \{1, \cdots, n\}$ such that $\mathbf{Prob}(\hat{S} = S) = p_s$
    update selected coordinates:$x^{k+1} = x^k - \sum_{i \in \hat{S}} \frac{1}{w_i} \nabla_i f(x) e^i$
end for

As we can see this algorithm is not parallel in a computing way, is parallel in the selection and update the coordinate since a set of coordinates are selected to update.

Inexact Coordinate Descent (ICD)

Tappenden et al. (2013) presented ICD. This work presented a block coordinate descendent method when a inexact update are applied. This algorithm in each iteration pick a blok of coordinates $i \in \{1, \cdots, n\}$ with probability p_i already defined, and updates them, regarding the "level of inexactness" established.

Lets consider

$$V_i(x, t) := \langle \nabla_i f(x), t \rangle + \frac{l_i}{2} \|t\|^2_{(i)} + \mathcal{R}_i(x^{(i)} + t) \tag{2.22}$$

This algorithm best applies to situations where is easier to approximately minimize $t \to V_i(x,t)$ than either to approximately minimize $t \to F(x+U_i t)$ and/or exactly minimize $t \to V_i(x,t)$.

In some cases calculate exact update is impossible, or is computational complex and infeasible in terms of time of execution. The propose of this algorithm was to allow *inexactness* in update step, in this way allowing a higher range of problems were *Coordinate Descendent* could be applied.

Semi-Stochastic Coordinate Descent (S2CD)

Konečný et al. (2014) presented this algorithm. In this work, the authors consider the strongly convexity of F and considered that is the average of multiple strongly convex functions. The algorithm is split into two different steps that are in a loop. First, they started with a deterministic calculation of the gradient followed by a stochastic step, this step is the point of divergence from other algorithms in the literature. In this step, the algorithm selects a function f_i and a coordinate j at random using nonuniform distributions and update a single coordinate. This is not a parallel algorithm since it only updates a single coordinate by iteration. The algorithm itself is represented as follows:

Algorithm 5 S2CD

Set m (max # of stochastic steps per epoch),$h > 0$ (stepsize parameter); $x_0 \in \mathbb{R}^n$ (starting point)
for $k = 0, 1, \cdots$ **do**
  compute and store $\nabla f(x_k) = \frac{1}{n}\sum_i \nabla f_i(x)$
  initialize inner loop: $y_{k,0} \leftarrow x_k$
  Let $t_k = T \in \{1, \cdots, n\}$ with probability $\frac{(1-\mu h)^{m-T}}{\beta}$
  for $t = 0$ to $t_k - 1$ **do**
    Pick coordinate $j \in \{1, \cdots, n\}$
    Pick function index i from the set $\left\{i : L_{ij} > 0\right\}$ with probability q_{ij}
    Update the j^{th} coordinate:
    $y_{k,t+1} \leftarrow y_{k,t} - h p_j^{-1}\left(\nabla_j f(x_k) + \frac{1}{n q_{ij}}\left(\nabla_j f_i(y_{k,t}) - \nabla_j f_i(x_k)\right)\right) e_j$
  end for
  Reset the starting point: $x_{k+1} \leftarrow y_{k,t_k}$
end for

Accelerated Parallel Proximal Coordinate Descent Method (APPROX)

This algorithm was presented by Fercoq and Richtárik (2013) and the proposed method is simultaneously Accelerated, Parallel and Proximal. This method was proposed for minimizing a sum of convex functions, where each one of them depends on a small set

of coordinates. This method is more capable to obtain a high-accuracy solution in non-strongly convex problems since it's accelerate, can take advantage of parallel computing due to its parallel nature, has a faster convergence rate since longer stepsizes are proposed and there is no need to perform full-dimensional vector operations. The stepsize proposed in this work is based on *ESO*, this framework was presented in Qu and Richtárik (2014b), in this work a new model of *ESO* was presented.

Algorithm 6 APPROX

Choose $x_0 \in \mathbb{R}^N$ and set $z_0 = x_0$ and $\theta_0 = \frac{\tau}{n}$
for $k \geq 0$ **do**
  $y_k = (1 - \theta_k)\, x_k + \theta_k z_k$
  Generate a random set of blocks $S_k \sim \hat{S}$
  $Z_{k+1} = z_k$
  for $i \in S_k$ **do**
    $Z_{k+1}^{(i)} = arg\ \min_{z \in \mathbb{R}^{N_i}} \left\{ \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k v_i}{2\tau} \left\| z - z_k^{(i)} \right\|_{(i)}^2 + \psi_i(z) \right\}$
  end for
  $x_{k+1} = y_k + \frac{n}{\tau}\theta_k\, (z_{k+1} - z_k)$
  $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
end for

ALPHA

Qu and Richtárik (2014a) proposed this method based on arbitrary sampling, that is a concept not yet studied in the previous literature. Basically, this method at each iteration selects a subset of coordinates at random, following an arbitrary distribution. This method is extraordinarily flexible since the authors proposed a general randomized Coordinate descendent method, that could be implemented in a serial or parallel way, with or without arbitrary sampling. They focused on non-strongly convex function study, and they refer that further studies must be performed in order to set a unified algorithm and complexity analysis for strongly and non-strongly convex functions.

Algorithm 7 ALPHA

Parameters: proper sampling $\hat{S}$ with probability vector $p = (p_1, \cdots, p_n)$,$v \in \mathbb{R}^n_{++}$, sequence $\{\theta_k\}_{k\geq 0}$
Initialization: choose $x_0 \in dom\psi$ and set $z_0 = x_0$
for $k \geq 0$ **do**
    $y_k = (1 - \theta_k)\, x_k + \theta_k z_k$
    Generate a random set of blocks $S_k \sim \hat{S}$
    $Z_{k+1} \leftarrow z_k$
    for $i \in S_k$ **do**
        $Z^i_{k+1} = arg\ \min_{z\in\mathbb{R}^{N_i}} \left\{ \langle \nabla_i f(y_k), z\rangle + \frac{\theta_k v_i}{2p_i} \left\| z - z^i_k \right\|^2_i + \psi^i(z) \right\}$
    end for
    $x_{k+1} = y_k + \theta_k p^{-1} \cdot (z_{k+1} - z_k)$
end for

Hydra^2

Hydra^2 was presented by Fercoq et al. (2014), this algorithm is an specialization of **APPROX** method with a distributed setting or an accelerate version of **Hydra** algorithm proposed by Richtárik and Takáč (2012). This algorithm conceptually was based on a computer network, and within each one of this computers, we take advantage in parallel computing recurring to each one of its cores. They also proposed a new stepsize parameters in order that D_{ii} satisfy **ESO** assumption. In their experiments, they showed that the new easily computable stepsize achieves a comparable convergence speed with respect to others. As we can see in the algorithm he can easily set this as a parallel algorithm by simple consider that ou computer network only has 1 machine. They also showed that even though the cost per iteration, in runtime, is higher in Hydra^2, than in Hydra, Hydra^2 converges faster than Hydra. The algorithm is represented as follows:

Algorithm 8 Hydra^2

Set $\{\mathcal{P}_l\}_{l=1}^{c}$, $1 \leq \tau s$,$\{\mathcal{D}_{ii}\}_{ii=1}^{d}$,$z_0 \in \mathbb{R}^d$
Set $\theta_0 = \frac{\tau}{s}$ and $u_0 = 0$
for $k \geq 0$ **do**
    $z_{k+1} \leftarrow z_k$, $u_{k+1} \leftarrow u_k$
    for each computer $l \in \{1, \cdots, c\}$**in parallel do**
        pick a random set of coordinates $\hat{S}_l \subseteq \mathcal{P}_l$, $\left|\hat{S}_l\right| = \tau$
        for each $i \in \hat{S}_l$**in parallel do**
            $t_k^i = arg\ \min_t f_i^{'}(\theta_k^2 u_k + z_k)t + \frac{s\theta_k \mathcal{D}_{ii}}{2\tau} t^2 + R_i(z_k^i + t)$
            $z_{k+1} \leftarrow z_k^i + t_k^i,\ u_k^i \leftarrow u_k^i - \left(\frac{1}{\theta_k^2} - \frac{s}{\tau\theta_k}\right) t_k^i$
        end for
    end for
    $\theta_{k+1} = \frac{1}{2}\left(\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2\right)$
end for
Output:$\theta_k^2 u_{k+1} + z_{k+1}$

ARock

Peng et al. (2015) presented a parallel algorithm that differs from others in its nature. The algorithms that we saw are *synchronous* parallel (sync-parallel), this type of computing all Agents (computers, cores, etc.) must wait for all of them have finished their execution, so the algorithm is as fast as it slowest agent. This paper proposes a *asynchronous* parallel (async-parallel) computing approach, this approach allows the agents to run continuously and spread communication and memory access. This approach allows the failure insertions of agents since they are capable of continuous running. But this approach brings a complication on ensuring the convergence.

In their work, the authors presented several versions and extensions of the algorithm based on the problem that they study.

2.6 Summary

The algorithms that we discussed previously are some of thus that are represented on the wide range of works in Convex optimization. This is some that incorporate some important features to this work, they are Stochastic, we choose at random a set of coordinates to update, they are (mostly) parallel, in the sense that we select more than one coordinate in each iteration.

The following table summarizes the algorithms that were presented in the previous sections, identifying if they are conceptualized to parallel computing (column "Parallel") and if it has been applied to *LASSO* (column "LASSO").

Name	Paper	Parallel	LASSO
RCDM	Nesterov (2010)		
Shotgun	Bradley et al. (2011)	✓	✓
GRock	Peng et al. (2013)	✓	✓
'NSync	Richtárik and Takác (2013)		
APPROX	Fercoq and Richtárik (2013)	✓	✓
ICD	Tappenden et al. (2013)		
S2CD	Konečný et al. (2014)		
APLHA	Qu and Richtárik (2014a)		
Hydra^2	Fercoq et al. (2014)	✓	✓
ARock	Peng et al. (2015)	✓	✓

Table 2.1: Algorithms in Literature

Chapter 3

Contributions to LASSO-VAR Model Fitting

This work intends to give an *LASSO* approach to the estimation of the coefficient matrix, B, of the VAR problem (2.9). This approach is commonly addressed as the ***LASSO-VAR***, This problem can be described as the following expression:

$$\min_{B} \quad \frac{1}{2} \|Y - BZ\|_F^2 + \lambda \|B\|_1 \tag{3.1}$$

To find an estimation $\hat{B}$ for the coefficient matrix B we will apply the algorithms **ShotGun** and **GRock** described in the former chapters. Both algorithms will be applied with the traditional sequential programming techniques and by using parallel computer programming techniques when these are possible.

Besides aforementioned changes, it terms of the LASSO-VAR problem it was also taken into consideration some assumptions. Considering a $VAR_k(p)$ problem, then a pth-order can be expressed as:

$$Y_t = v + \sum_{l=1}^{p} B_l Y_{t-l} + u_t \tag{3.2}$$

where $Y_t, v, u_t \in \mathbb{R}^k$ each B_l represents a coefficient matrix $k \times k$. We can translate equation (3.2) to a matrix notation as

$$Y = v1^{'} + BZ + U \tag{3.3}$$

Where 1 denotes the vectors of ones, with dimension $T \times 1$, and v the intercept vector We intend to estimate B applying LASSO strategy, so by extending Tibshirani (1994) we get the following objective function:

$$\frac{1}{2} \left\|Y - v1^{'} - BZ\right\|_F^2 + \lambda \|B\|_1 \tag{3.4}$$

In order to simplifying computation we might consider the centred form, so equation

(3.4) becomes

$$\frac{1}{2}\left\|\mathbf{Y} - B\mathbf{Z}\right\|_F^2 + \lambda\left\|B\right\|_1 \tag{3.5}$$

where $\mathbf{Y} = Y - \bar{Y}$ and $\mathbf{Z} = Z - \bar{Z}$

So with all of this in consideration we can establish that the v Intercept term can be expressed by the following equation:

$$v = \bar{Y} + \hat{B}\bar{Z} \tag{3.6}$$

We already know that LASSO is given by $F(B) = G(B) + \lambda\mathbf{R}(B)$, where $\mathbf{R}(B)$ is not differentiable we must consider sub-gradient of $F(B)$, so:

$$\nabla F(B) = \nabla G(B) + \psi(R(B)) \tag{3.7}$$

where

$$\psi(R(B)) \in \begin{cases} \{sgn(R(B))\} & , R(B) \neq 0 \\ [-1, 1] & , R(B) = 0 \end{cases} \tag{3.8}$$

3.1 Algorithms

As presented in Bradley et al. (2011) and Peng et al. (2013), the algorithms were studied to a vector lasso estimation. In the article they solved lasso estimation $\hat{x}$ for the problem:

$$\min_x \quad \frac{1}{2}\left\|y - Ax\right\|_2^2 + \lambda\left\|x\right\|_1 \tag{3.9}$$

Our problem is the estimation of the matrix B, so we need to tweak a little the algorithms to be able to apply them to the Lasso-VAR problem.

There were no majors changes to the algorithms, these changes were in the selection criteria as well to the gradients. Both algorithms stands under the assumptions of Convex Optimization algorithms for problems:

$$\min \quad F(B) = G(B) + \lambda R(B) \tag{3.10}$$

were G is differentiable and R is separable. So as we can see in equation (3.1) we can describe $G(B) = \frac{1}{2}\left\|Y - BZ\right\|_F^2$ and $R(B) = \left\|B\right\|_1$, then both algorithms will use the gradient of $G(B)$ that is $\nabla G(B) = -(Y - BZ)Z^T$. Taking all this in consideration both algorithms will aplly a Soft-Threshold to the update, we will consider the generic equation for this soft-threshold as:

$$\mathcal{ST}_{\lambda,\rho}(x) = sign(x)max\{|x| - \lambda\rho, 0\} \tag{3.11}$$

Besides the programming paradigm, we will also see this as a single problem as well as sub-problems. This means that we will apply both algorithms and programming paradigms to the problem (3.1) as well to each sub-problem to each location, solving all the 44 sub-problems of estimation $\hat{B}$ by estimate each line as a single problem.

3.1.1 Shotgun

This algorithm is described in algorithm 2, as discussed above this algorithm will be changed as explained in the following sub-sections.

But before we address the algorithm changes lets take a look at the step-size condition for this algorithm. As is mentioned in Bradley et al. (2011), the convergence is directly related to the step-size update. This step-size assures the algorithm convergence, but this is a throwback since that for a better convergence implies a higher runtime of the algorithm, this means that for a better solution the model training time is higher. For this work, we used the step-size proposed in Nicholson et al. (2014). In this article, the authors proposed a step-size given by :

$$\rho = \frac{1}{\lambda_{max}} \tag{3.12}$$

Where λ_{max} is the higher eigenvalue of ZZ^T. This operation as also a considerable runtime and therefore increase the model training.

Sub-Problems

To this approach we need to consider the algorithm 2. The main difference in this approach is a new cycle in order to run it for all sub-problems an in the computation of the gradient. To the sub-problem approach the gradient is given by:

$$\nabla f(b) = -Z^T(y - Zb) \tag{3.13}$$

Where y represents the current sub-problem i, $y = Y_{(i)}$ and b represents the line from B that we intend to estimate in each sub-problem. So the algorithm that was applied is:

Algorithm 9 Shotgun Sub-problems

choose $P \geq 1$, the number of parallel updates
Set $B = 0_{k \times kp}$, ϵ and λ
for each $j \in \{1, \cdots, k\}$ **do**
    $b \leftarrow B[j,] = 0_{1 \times kp}$
    $y \leftarrow Y[j,]$
    while threshold>ϵ **do**
        $b^{old} \leftarrow b$
        Choose P positions uniformly at random $r \in \{1, \cdots, k\}$,
        Compute $g_{j,r} := \nabla f(b^{old})_r$
        Update $b_r \leftarrow \mathcal{ST}_{\lambda\rho}(b_r^{old} - \rho g_r)$
        update threshold
    end while
end for
compute (3.6) to estimate $\hat{v}$

As we you can see there is no major changes to this algorithm

VAR

This approach leads similar changes, in this case, it will not have this new cycle as in the sub-problems, but instead of the selection is a position in a vector we will have a selection in a matrix. This selection is somehow different from the original algorithm. In a matrix, we need two coordinates to set a coordinate to update, in terms of the algorithm we will have two random selectors. And besides this change, as in the sub-problem approach, the computation of the gradient is also different. I this case it will be applied $\nabla G(B) = -(Y - BZ)Z^T$, hence the algorithm can be described as:

Algorithm 10 Shotgun VAR

choose $P \geq 1$, the number of parallel updates
while threshold>ϵ **do**
    $B^{old} = B$
    Choose uniformly at random $j \in \{1, \cdots, k\}$ *and* $r \in \{1, \cdots, kp\}$
    Compute $G_{j,r} := \nabla G(B^{old})_{j,r}$
    Update $B_{j,r} \leftarrow \mathcal{ST}_{\lambda\rho}(B_{j,r}^{old} - \rho G_{j,r})$
    update threshold
end while
compute (3.6) to estimate $\hat{v}$

3.1.2 GRock

As presented in Peng et al. (2013), this algorithm stands under the condition that matrix Z has a $2-norm$ column value, this will be the main condition to determinate the updated merit as described in the algorithm 3 (the original algorithm presented on Peng et al. (2013)). In this algorithm, only the P coordinates with higher merit will be chosen to be updated. Equations (2.20) and (2.21) described the merit as well the update selection of original algorithm. In this work, this assumptions remain valid, but the merit calculation will be slightly different. The merit will use the $2-norm$ assumption and use this $2-norm$ column values to determinate the merit of each update.

The process to determinate the updated merit will be described next, first, let's consider

$$Z_{2-norm} = \left[\left\|Z_{[1]}\right\|_2^2 \ \left\|Z_{[2]}\right\|_2^2 \ \left\|Z_{[3]}\right\|_2^2 \ \cdots \ \left\|Z_{[kp]}\right\|_2^2 \right] \tag{3.14}$$

the vector that represents the $2-norm$ column of Z. After this vector is calculated we can consider the β to being:

$$\beta = B - \frac{\nabla G(B)}{Z_{2-norm}} \tag{3.15}$$

After the β value is determinate we can apply a *soft-threshold* to it, being this given by equation (3.16)

$$\beta_{threshold} = sign(\beta) max\left\{|\beta| - \frac{\lambda}{Z_{2-norm}}, 0\right\} \tag{3.16}$$

Then we consider d matrix as:

$$d = \beta_{threshold} - \beta \tag{3.17}$$

This d matrix is the matrix used to determinate the P updates with higher merit, therefore select to update.

With all this in consideration, we can now take a look at the changes of the algorithm, the aforementioned process to merit calculation is the major adaptation to the algorithm.

Sub-Problems

Since we are looking at each problem separately, the obvious change, just like in **Shotgun** algorithm, is the additional cycle to run each one of the locations. The merit calculation was already described. So the applied algorithm can be described as:

Algorithm 11 GRock Sub-problems

choose $P \geq 1$, the number of parallel updates
Set $B = 0_{k \times kp}$, ϵ and λ
compute Z_{2-norm}, as described in (3.14)
for each $j \in \{1, \cdots, k\}$ **do**
    $b \leftarrow B[j,] = 0_{1 \times kp}$
    $y \leftarrow Y[j,]$
    while threshold>ϵ **do**
        $b^{old} \leftarrow b$
        compute β, as described in (3.15)
        compute d, as described in (3.17)
        Select P coordinates with higher update merit.
        update $b_j \leftarrow b_j + d\,j$
        update threshold
    end while
end for
compute (3.6) to estimate $\hat{v}$

VAR

For this approach, there is no major besides of the merit calculation. After it is only the position of those in the matrix that is calculated to be updated. So the algorithm is quite similar to the previous minus the obvious cycle to run each sub-problem. Hence the algorithm is:

Algorithm 12 GRock Sub-problems

choose $P \geq 1$, the number of parallel updates
Set $B = 0_{k \times kp}$, ϵ and λ
compute Z_{2-norm}, as described in (3.14)
while threshold>ϵ **do**
    compute β, as described in (3.15)
    compute d, as described in (3.17)
    Select P coordinates with higher update merit.
    update $B_{j,r} \leftarrow B_{j,r} + d_{j,r}$
    update threshold
end while
compute (3.6) to estimate $\hat{v}$

3.2 Summary

Let's recap, just a few concepts that we saw till now. In this work, we will consider 2 major approaches to the problem, consider the problem as a whole and apply **LASSO-VAR** to estimate the coefficient matrix $\hat{B}$. And another approach that is splitting the problem to each one of the location, meaning that it will estimate for each location is the respective coefficient line.

Besides the approaches to the problem, we will also consider two different approaches in terms of computer programming paradigms. In terms of computer programming, it will consider the traditional sequential paradigm and also the parallel paradigm. In terms of algorithms that were presented the only change that is made is the P value, that in the sequential paradigm it will be 1, meaning that at each iteration only 1 update will be performed.

Chapter 4

Case Study

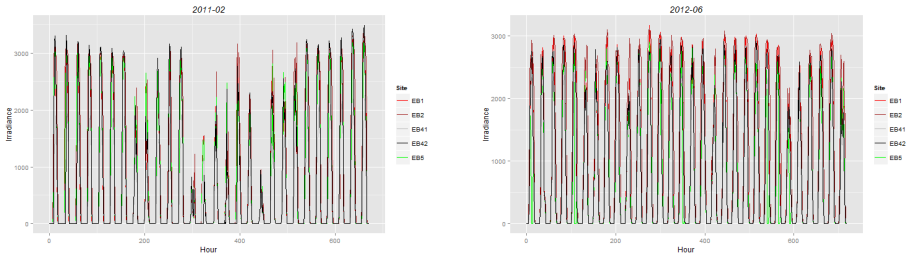
This chapter aims to present and analyse some results from the trained models. This results will be compared to the results obtained in Trindade (2014). In that work, the author applied an OLS estimator for $\hat{B}$. In this chapter, it will be presented the forecast models'error so that we can make a comparison in terms of their error and will be presented, as well, some forecasts for these models.

This chapter will be divided into two major sections according to the problem approach, **Sub-problem** and **VAR**, to the algorithms. In each of this section, the aforementioned process of error and forecast presentation will be applied.

But first, let's take a look at the used data for training and to test all models.

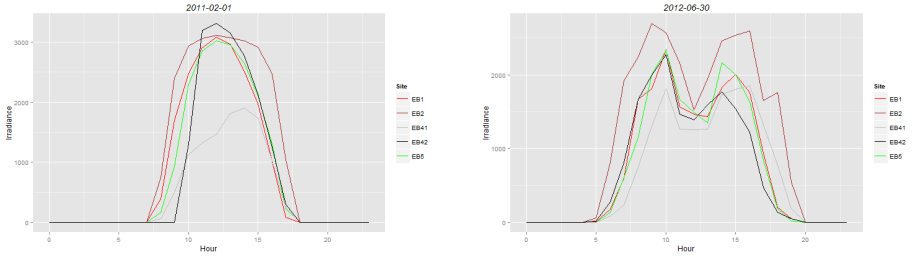
4.1 Data

This section aims to explain the used data to train and test the models of prediction, for the problem (3.1). The *PV* production is highly seasonal, the *PV* production is higher in the summer than it is in winter. Besides this seasonality of the seasons, this production is also dependent on the period of the day. On figure 4.1 we can see the difference in monthly production between winter month and summer month and in the figure 4.2 we can see the difference in daily production between winter day and summer day. We can see in 4.1 the behaviour of PV production in different months and seasons, in 4.1a we have PV production in a Winter time and on 4.1b we have the production on Summer. In 4.2 is represented a daily production on this months. As we can observe there are periods where there is no production, and that this is also affected by the season as expected since the solar exposition is larger at summer than at the winter.



(a) February of 2011 (b) June of 2012

Figure 4.1: Monthly Production



(a) February 1^{st}, 2011 (b) June 30^{th} 2012

Figure 4.2: Monthly Production

The used data to train and test all models are referent to 44 $one - hour$ time steps EB. Those EB's are part of 60 different $one - hour$ time step and represents a $1500km^2$ SMART Grid. Each one of those EB's is represented by 1082 observations made between $February\ 1^{st}$, 2011 and $March\ 6^{th}$, 2013. In a PV production, there is a time frame where there is no production made, this time frame is between 20 p.m. and 6 a.m. Due to this lack of production and since the models considers p_t and p_{t-1} as an input variable, the time frame for predictions are restricted depending on the time step. For example for $one - hour - ahead$ it is only possible to forecast for the period 9 a.m. to 19 p.m., and for $6 - hour - ahead$ forecast the time frame for that forecasting is 14.pm to 19 p.m. The record values were normalised using the Clear Sky Model that was explained on section 2.1. The Clear Sky Model is given by:

$$p_t^{norm} = \frac{p_t}{\hat{p}_t^{cs}} \tag{4.1}$$

On Trindade (2014) it was applied two different persistent models, those models instead of consider the prediction the record values they consider the forecast result from a$two - step$ process, This means that it take p_t^{norm} or p_{t-24+k}^{norm} as the normalised forecast values. Consider the models $persistent(\tau)$ being $\hat{p}_{t+k}^{norm} = p_t^{norm}$ and the model $persistent(\tau 24)$ as being $\hat{p}_{t+k}^{norm} = p_{t-24+k}^{norm}$. Then the forecast value is given by:

$$\hat{p}_{t+k} = \hat{p}^{norm}_{t+k} \hat{p}^{cs}_{t+k} \tag{4.2}$$

In this Work the clear sky parameters that was identified in Trindade (2014) $\tau = 0.85$ and for v we consider $v = 0.01$.

In the same work, the author presents the normalised methods to evaluate the models. The Root Mean Square is given by equation (4.3).

$$nRMS\,E_k = \frac{\frac{1}{N}\sqrt{\sum_{t=1}^{N}(p_{t+k} - \hat{p}_{t+k})^2}}{max(p_{t+k})} \tag{4.3}$$

To understand the model bias it is necessary the normalization of the bias as in equation (4.4).

$$nBias = \frac{\frac{1}{N}\sum_{t=1}^{N}(p_{t+k} - \hat{p}_{t+k})^2}{max(p_{t+k})} \tag{4.4}$$

As already described, in this work we intend to estimate $\hat{B}$ of the Lasso-VAR problem of equation (3.1). This issue will be addressed by two different programming paradigms, the traditional sequential method and in a parallel paradigm. First, lets take a look at the algorithms and the adaptations made in this work.

4.2 Sub-problem

In this sections, we will analyse the models obtained from the sub-problem approach to the problem. It will be presented for both algorithms.

4.2.1 RMSE

Figures 4.3 and B.1 represent the RMSE for step $t + 1$ for the parallel and sequential approach respectively. As we can see, the sequential technique does not bring any advantage and in terms of forecast error the Shotgun models are very similar to the OLS model, but the runtime for these models train is very expensive, so there is no gain in considering these models. In the other hand, the Shotgun parallel models are quite a bit better and the error of those models is lower than the OLS model error, even in the outlier stations the error is considerably lower than the OLS model. All models present similar errors but taking an overall look at those errors, model $\lambda = 0$ is lower error rate on $\approx 43\%$ stations.

The greater gain in these Shotgun models is not for step $t + 1$ but for the next steps. If we look at figure 4.4 we can see that the OLS model presents a significantly higher error than the Shotgun models. In these cases, even the sequential approach performs better

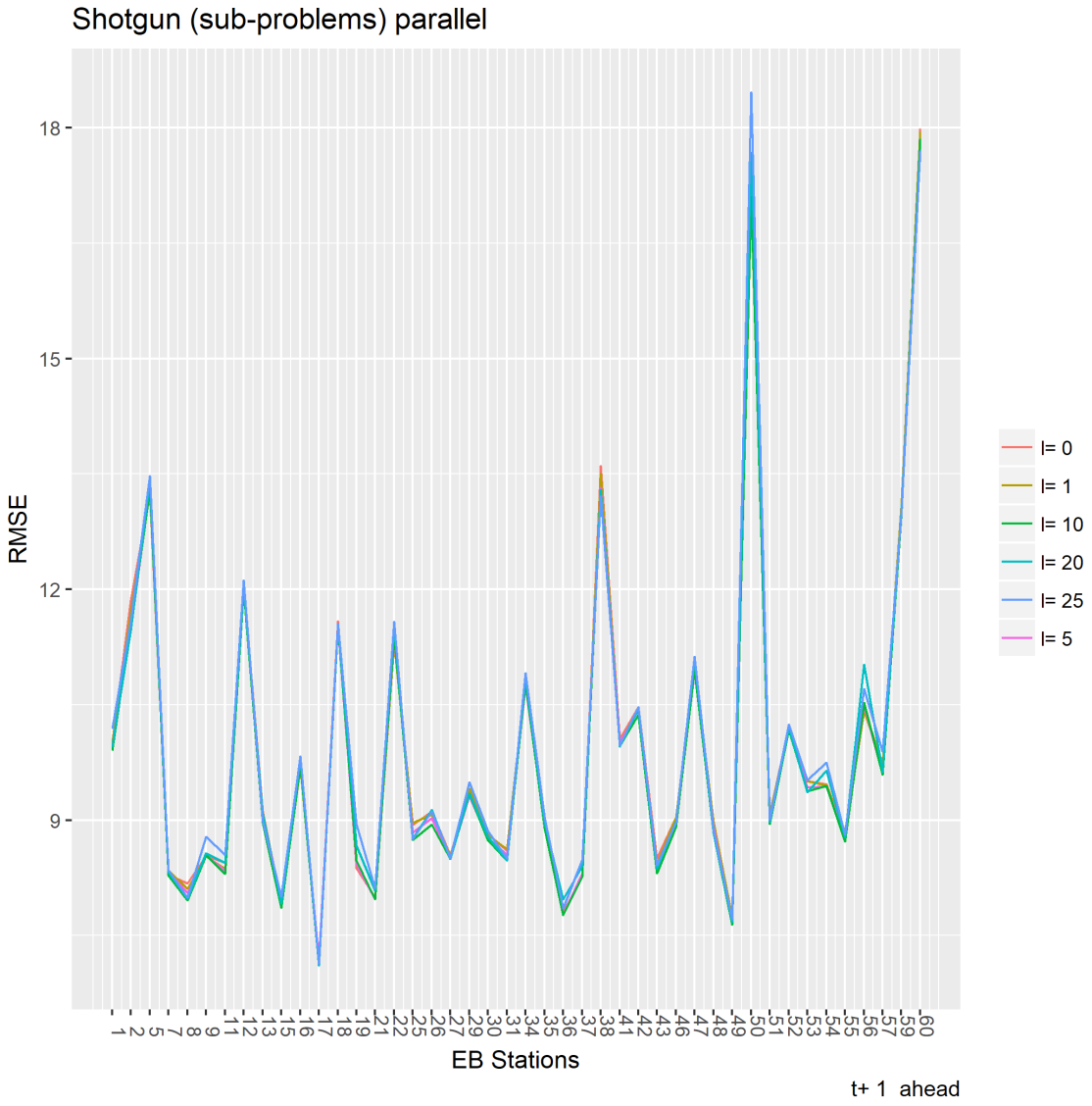


Figure 4.3: Shotgun (sub-problem) parallel models RMSE for $t + 1$

than the OLS model, even though they are worst than the parallel methods as we can see in the figure B.2 and the runtime is higher, so there is no gain in considering these models.

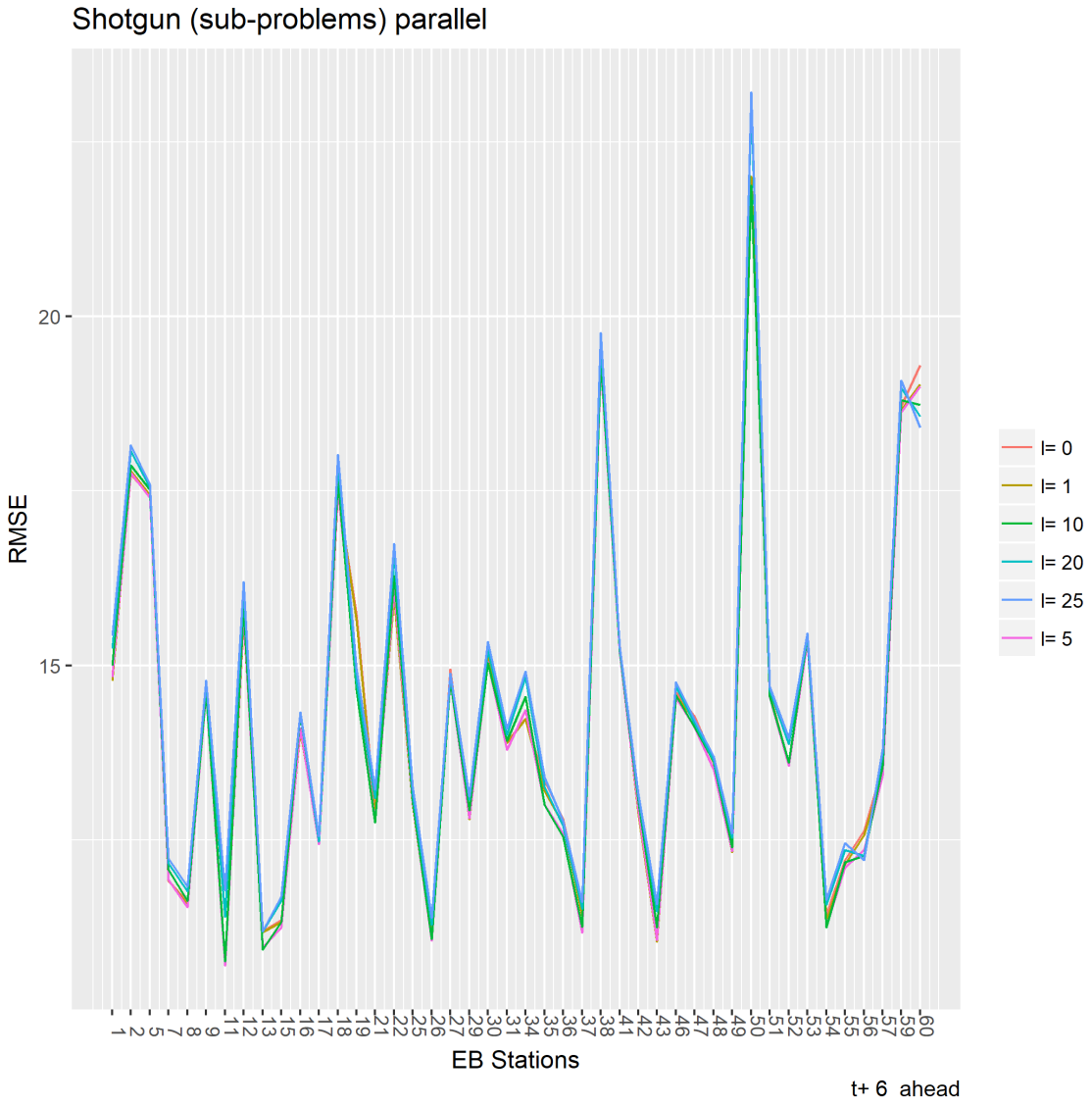


Figure 4.4: Shotgun (sub-problem) parallel models RMSE for $t + 6$

We saw the models for the Shotgun algorithm, let's now consider the models for GRock, with the same comparison in consideration.

We can see on figure 4.5 that the GRock model error is quite similar to the error obtained by the OLS model. In figure B.3 it is presented the performance obtained by the sequential methods and as we can see the performance of these models is quite worst, so the sequential method models are not one to consider.

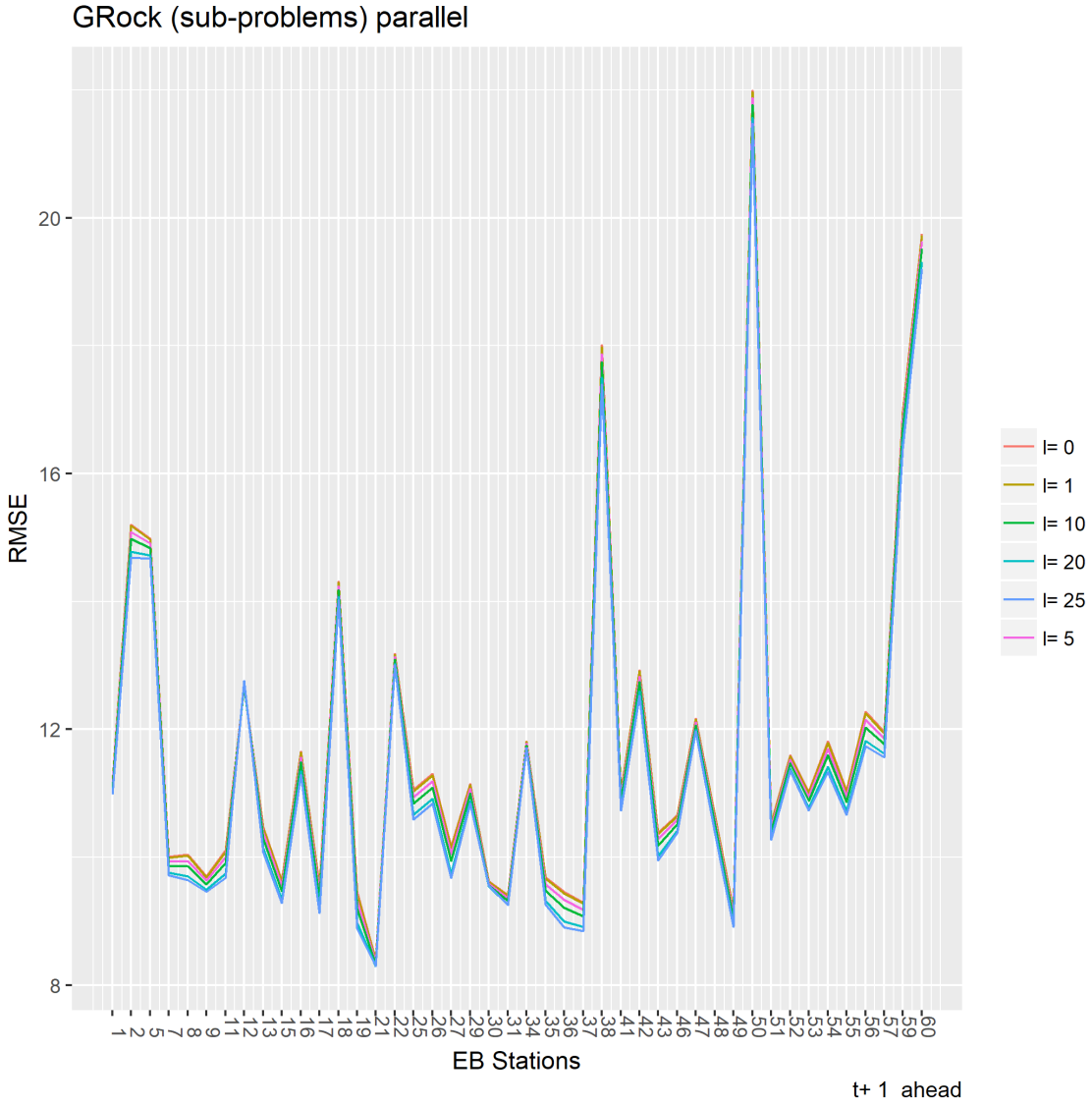


Figure 4.5: GRock (sub-problem) parallel models RMSE for $t + 1$

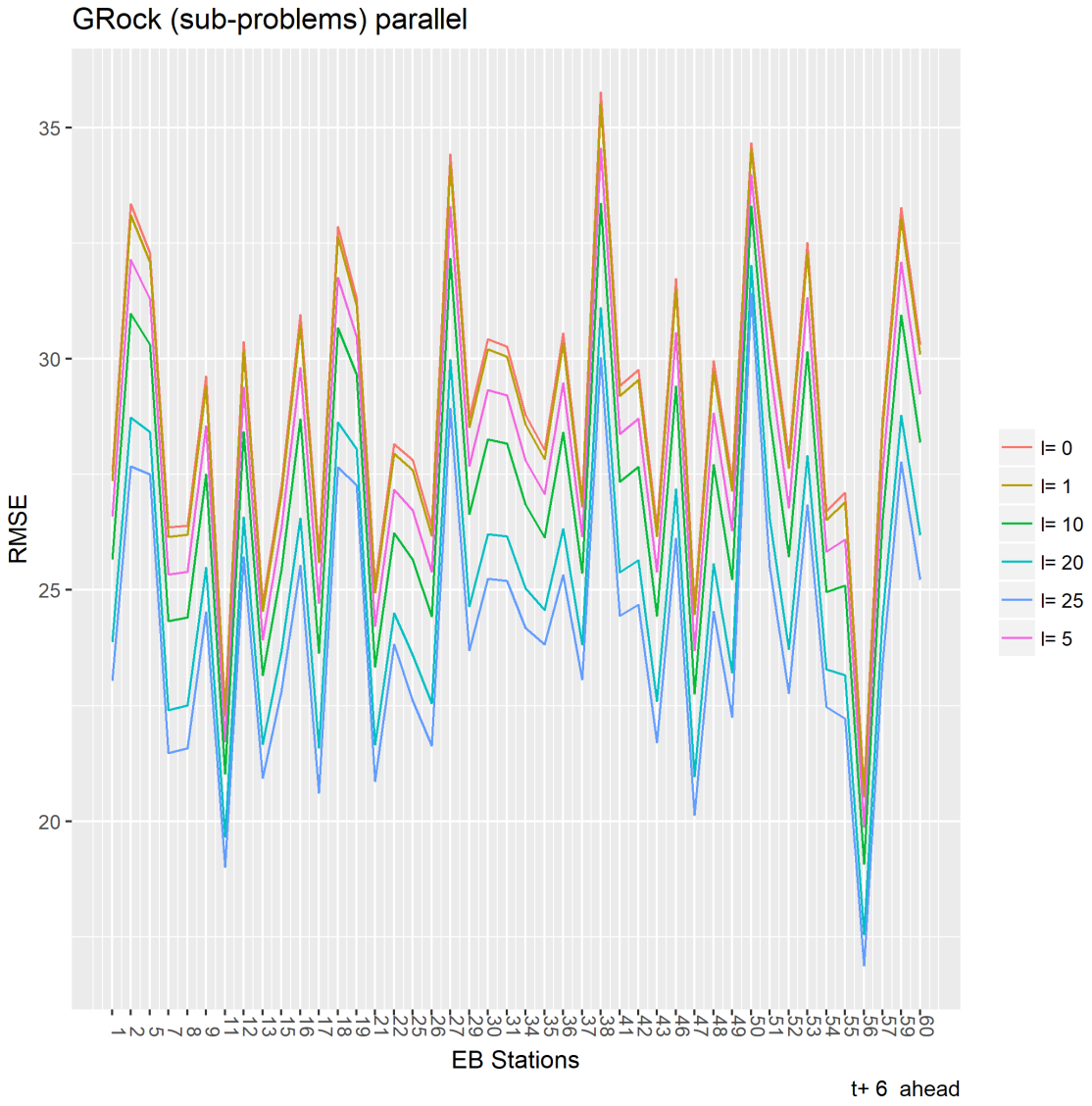


Figure 4.6: GRock (sub-problem) parallel models RMSE for $t + 6$

Unlike the Shotgun models, the GRock Models has no greater gain in the further steps, in figure 4.6 we can see that there are some stations where the error is lower by the GRock models and in the others the OLS is quite similar to the GRock models, but the difference is not significant. The same happens on the sequential methods that we can see on figure B.4. These results are quite better than the parallel methods, yet the difference is not enough to consider them.

4.2.2 Forecast

In this section we will take a look at the model's forecast, taking in consideration the same principal consideration of comparison with the OLS model. We will see some seasons'forecast (Spring, Summer, Autumn, and winter) to see the models'adaptation the different clouds formation as well as the time frame available for the forecast.

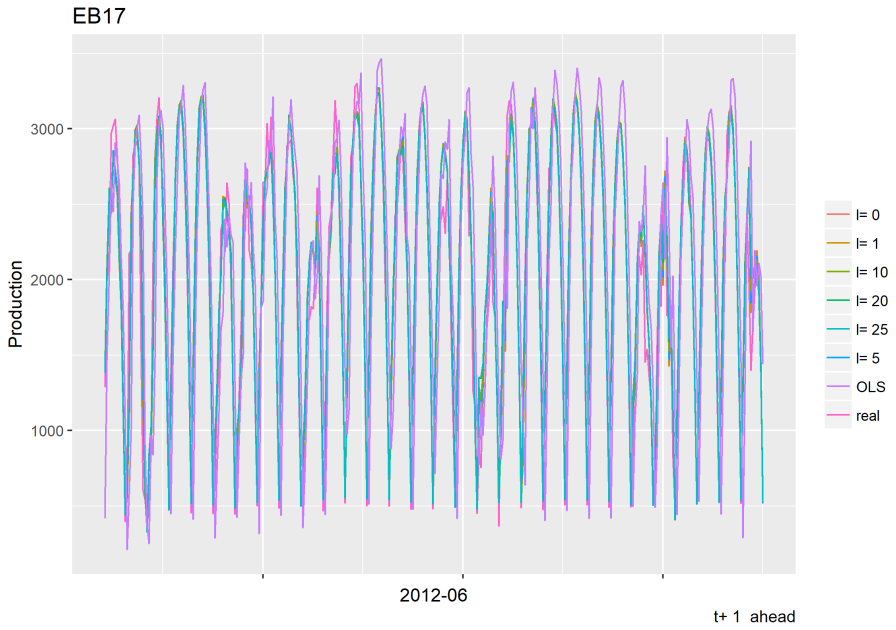


Figure 4.7: EB17 Shotgun (sub-problem) parallel models'step $t + 1$ forecast for 2012-06

As we can see on figure 4.7, the OLS model over-forecast in the major cases, meaning that the forecasts of this models usually is higher than the observed leading to an expectation that after will not correspond. Shotgun models are under-forecast those values, meaning that usually, these models are forecasting below the observed values.

As we can see on figure 4.8 the Shotgun models are better fit to the observed values, when compared with the OLS model. We can see that these models are more similar to the observed values than the OLS model. Considering figure 4.2b we can see that June 30^{th} 2012 as a cloudy day and the PV production ha some kind of "interruption", so lets consider the period 2012-06-28 until 2012-07-02. As we can see on figure 4.9 the shotgun models are quite fit to the observed values, on June 30^{th} the models does not fit as in the remain days, particularly between 12:00-15:00, at 15:00 the model and the observed are quite similar but after 15:00 the models starts do deviate again.

For step $t + 6$ these fit between shotgun models and observed values is also verified, yet for the cloudy day June 30^{th} this fit is not verified as it was somehow in step $t + 1$.

If we consider now the GRock models, we can see on figure 4.11 that these models

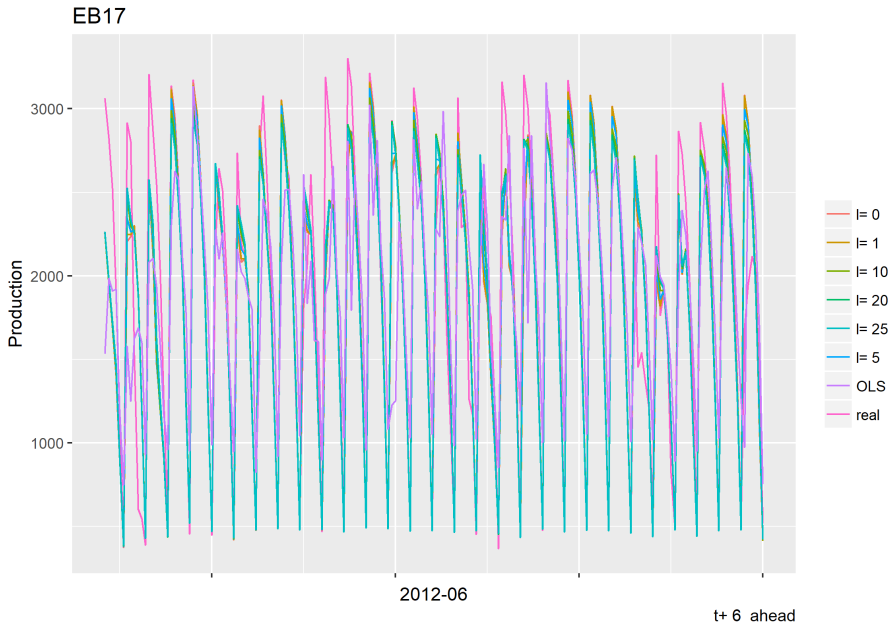


Figure 4.8: EB17 Shotgun (sub-problem) parallel models'step $t + 6$ forecast for 2012-06

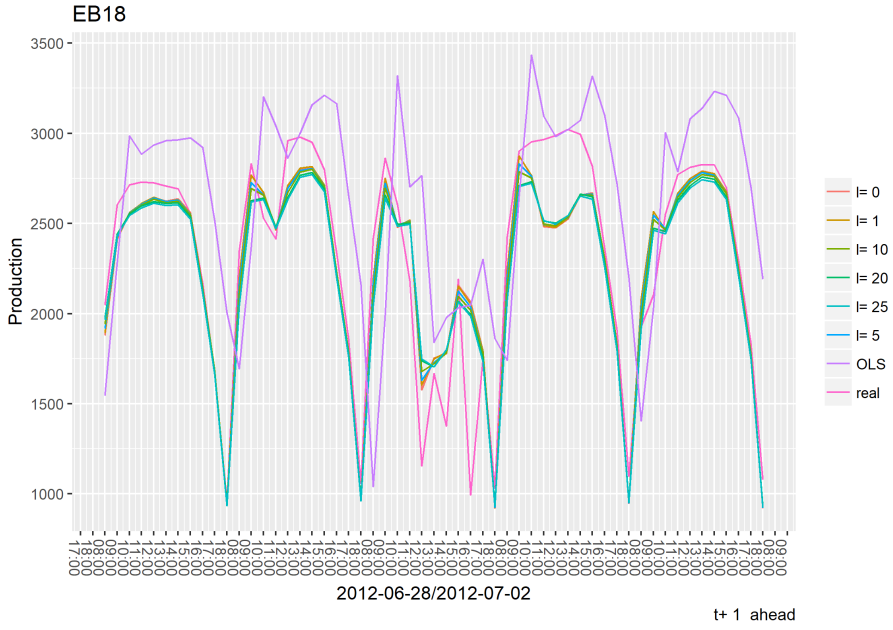


Figure 4.9: EB18 Shotgun (sub-problem) parallel models'step $t + 1$ forecast for period 2012-06-28/2012-07-02

are quite similar to the OLS models when considering step $t + 1$, but when considering step $t + 6$ start to deviate from it as we can see on figure 4.12.

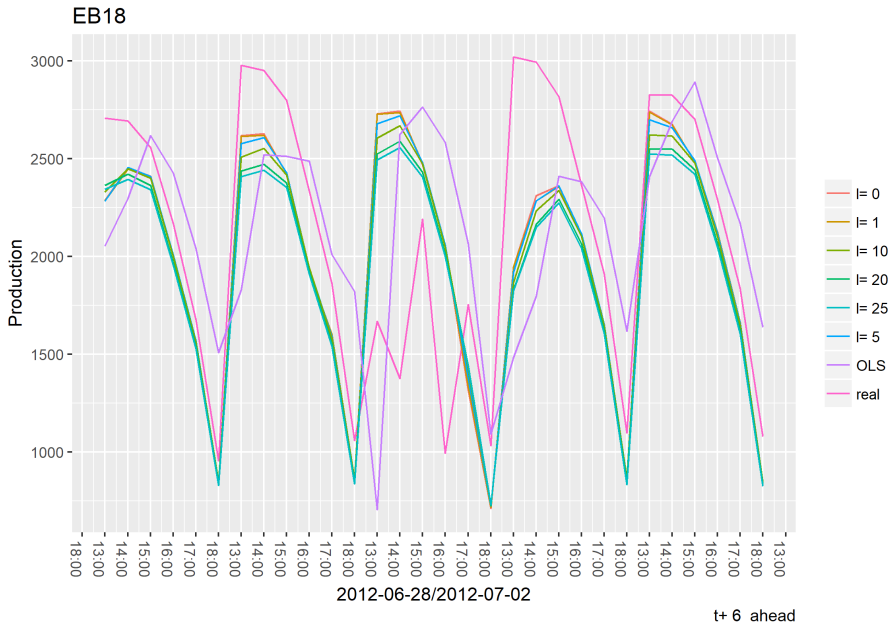


Figure 4.10: EB18 Shotgun (sub-problem) parallel models'step $t + 6$ forecast for period 2012-06-28/012-07-02

If we zoom in into the first 3 days (2012-02-02 until 2012-02-04) then we can see the that the GRock performs a little bit better than the OLS model. If we look at figure 4.13 we see that the models are very similar. A positive point of these models is that the trajectories of the forecast are very similar to the trajectories of the observed values, this feature is quite interesting. In figure 4.14 we can see that the GRock models are very deviated from the real value as well from the OLS model.

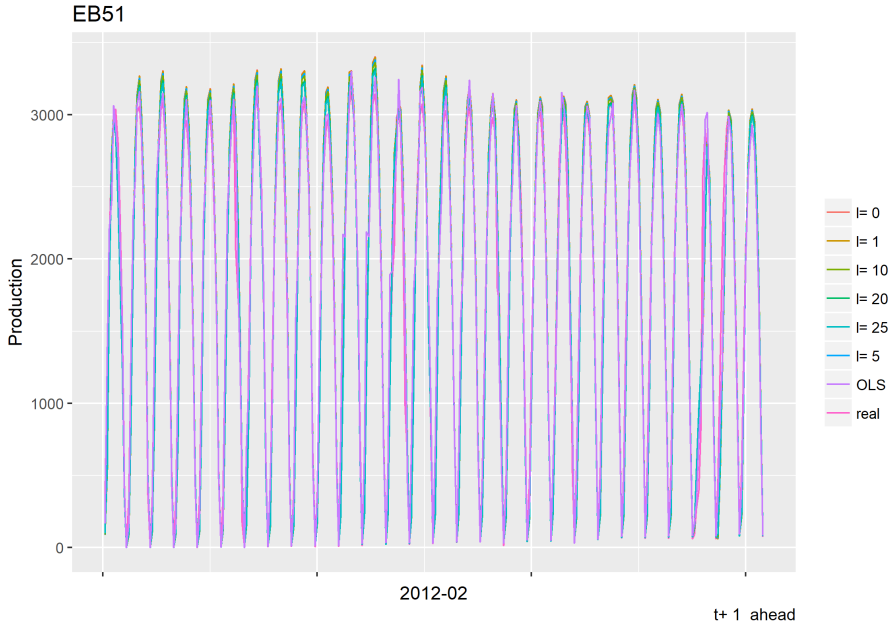


Figure 4.11: EB51 GRock (sub-problem) parallel models'step $t + 1$ forecast for 2012-02

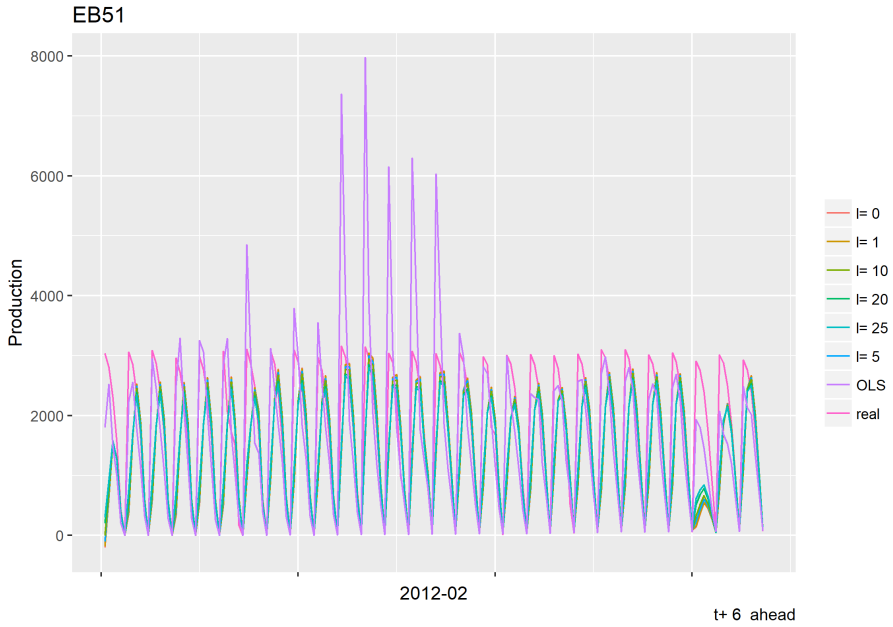


Figure 4.12: EB51 GRock (sub-problem) parallel models'step $t + 6$ forecast for 2012-02

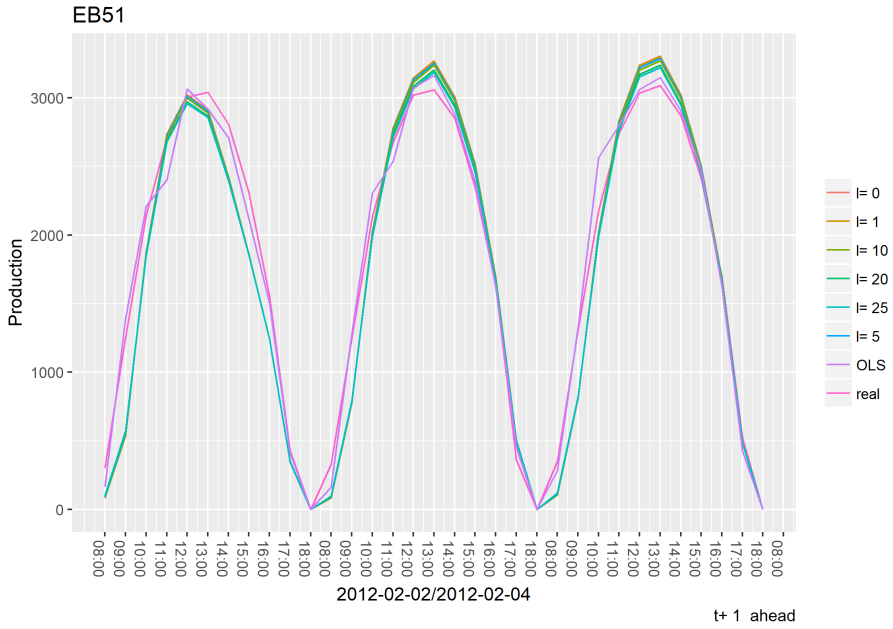


Figure 4.13: EB51 GRock (sub-problem) parallel models'step $t + 1$ forecast for period 2012-02-02/012-02-04

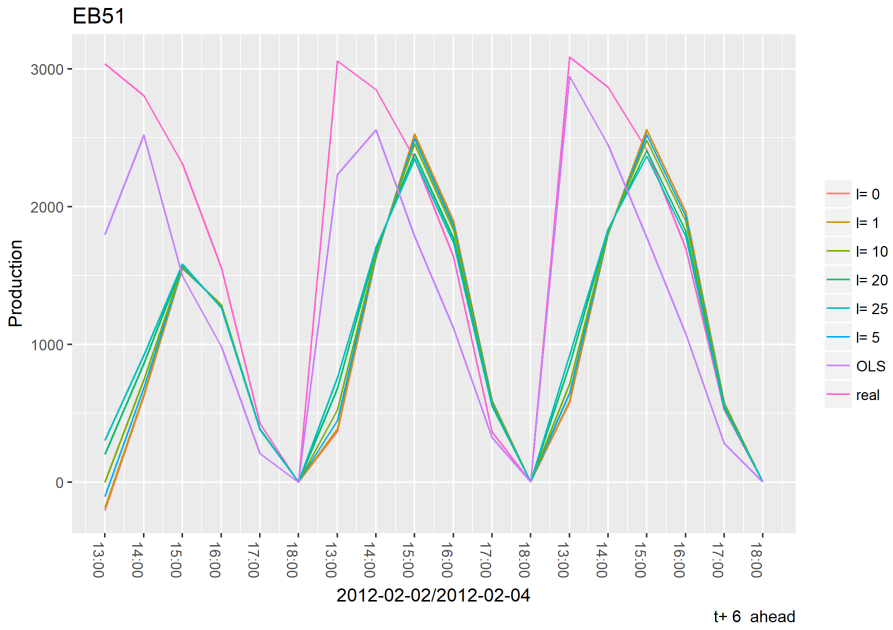


Figure 4.14: EB51 GRock (sub-problem) parallel models'step $t + 6$ forecast for period 2012-02-01/012-02-03

4.3 VAR

Let's now take a look at the problem approach, where the problem is considered as a whole.

4.3.1 RMSE

The Shotgun models'errors are presented in the following figures 4.15, 4.16, 4.17 and 4.18. Lets analyse them, as we can see this models follows the same path from the sub-problems'models. Meaning that these models bring a greater gain in the further steps, for step $t + 1$, in figures 4.15 and 4.16 it is possible to see a gain although these gains is are not so significant as the gains that obtained in the $t + 6$ models, in figures 4.17 and 4.18.

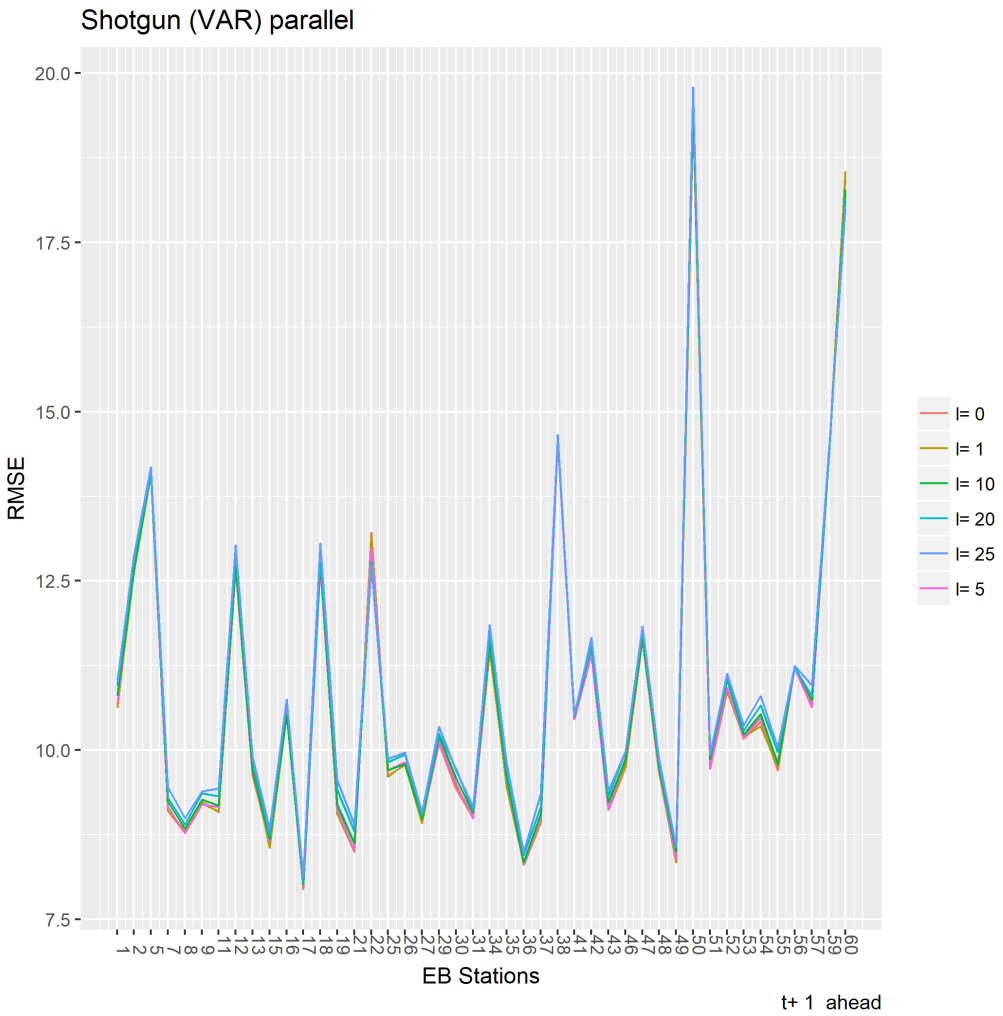


Figure 4.15: Shotgun (VAR) parallel models RMSE for $t + 1$

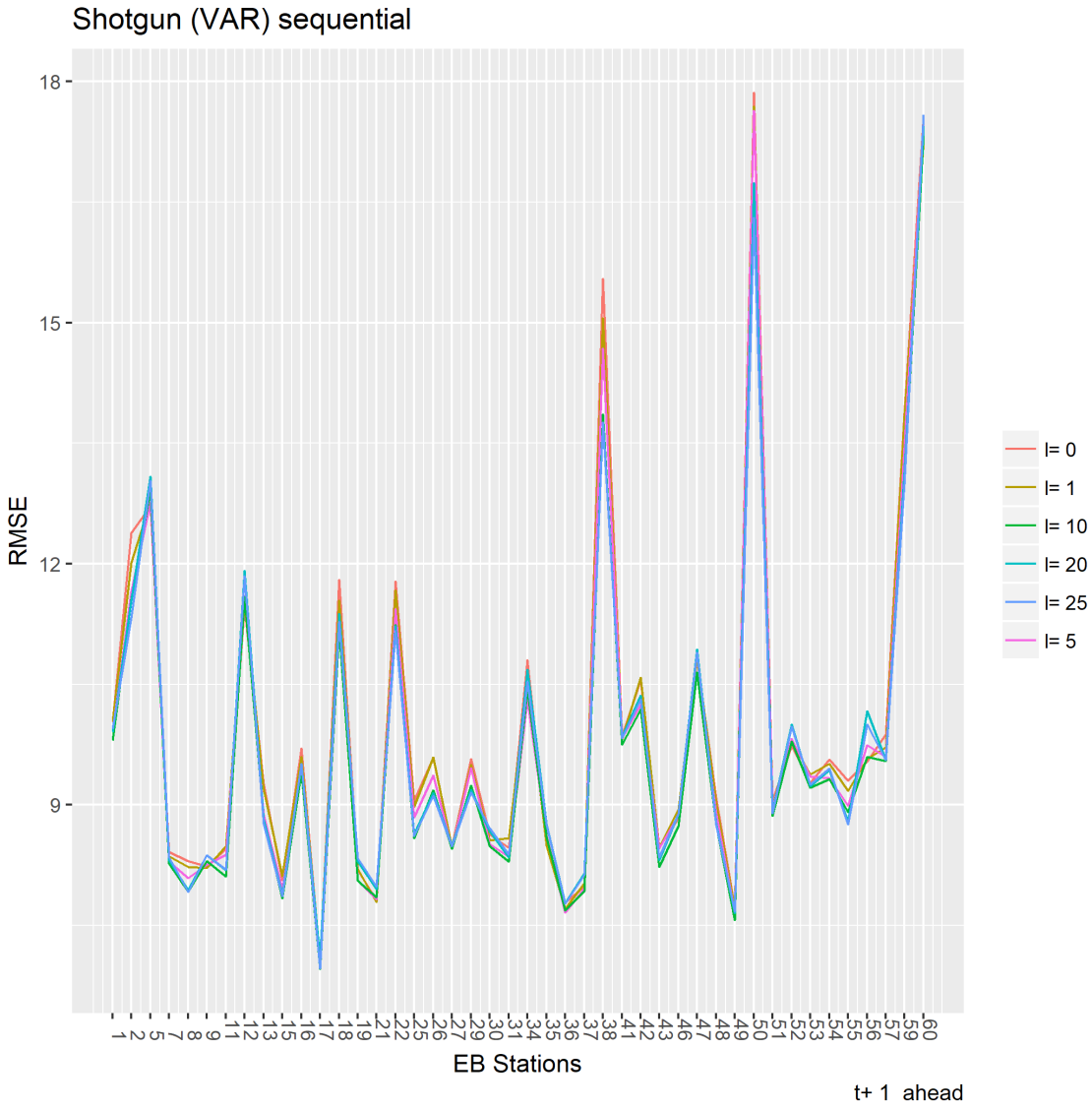


Figure 4.16: Shotgun (VAR) sequential models RMSE for $t + 1$

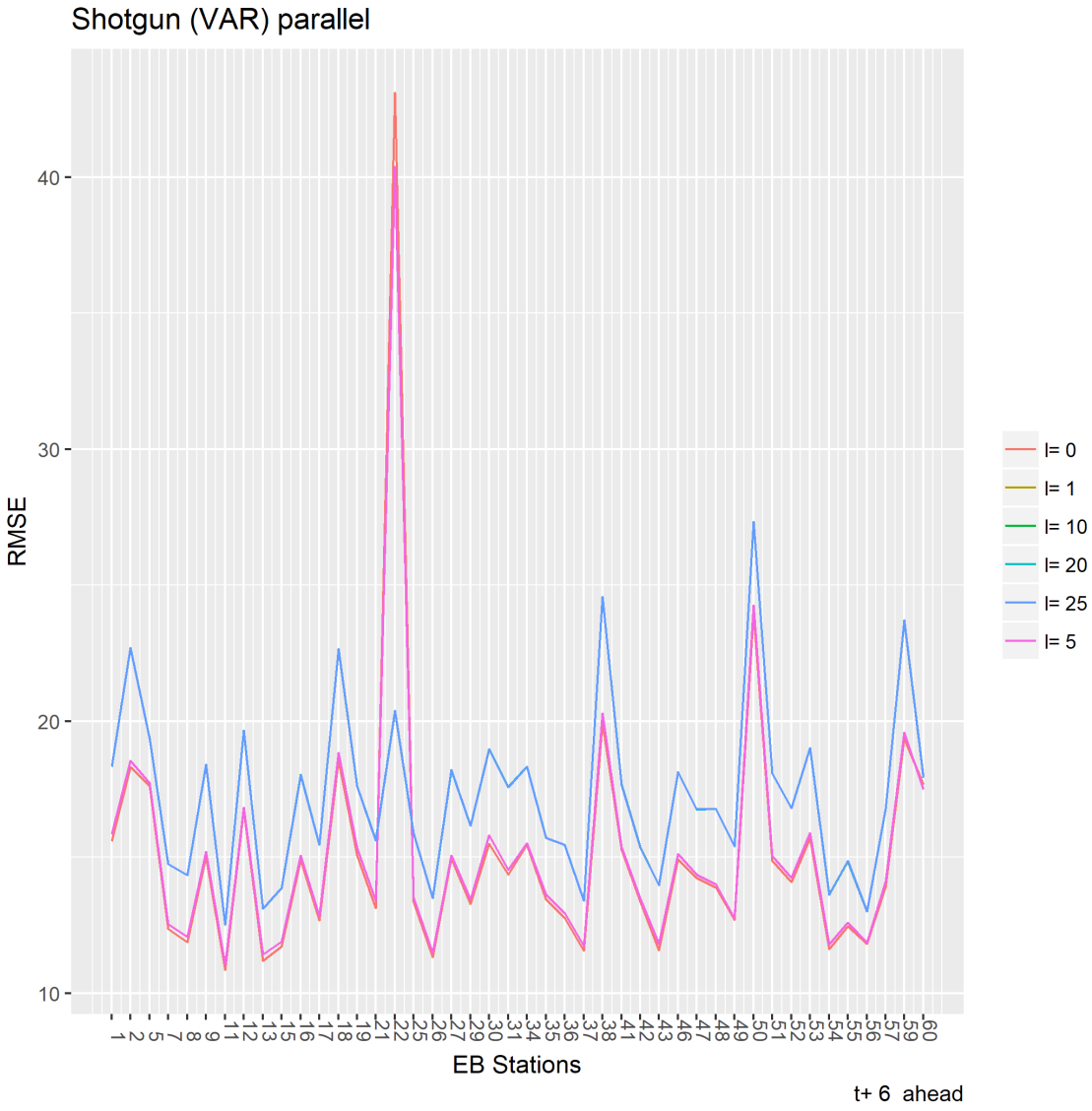


Figure 4.17: Shotgun (VAR) parallel models RMSE for $t + 6$

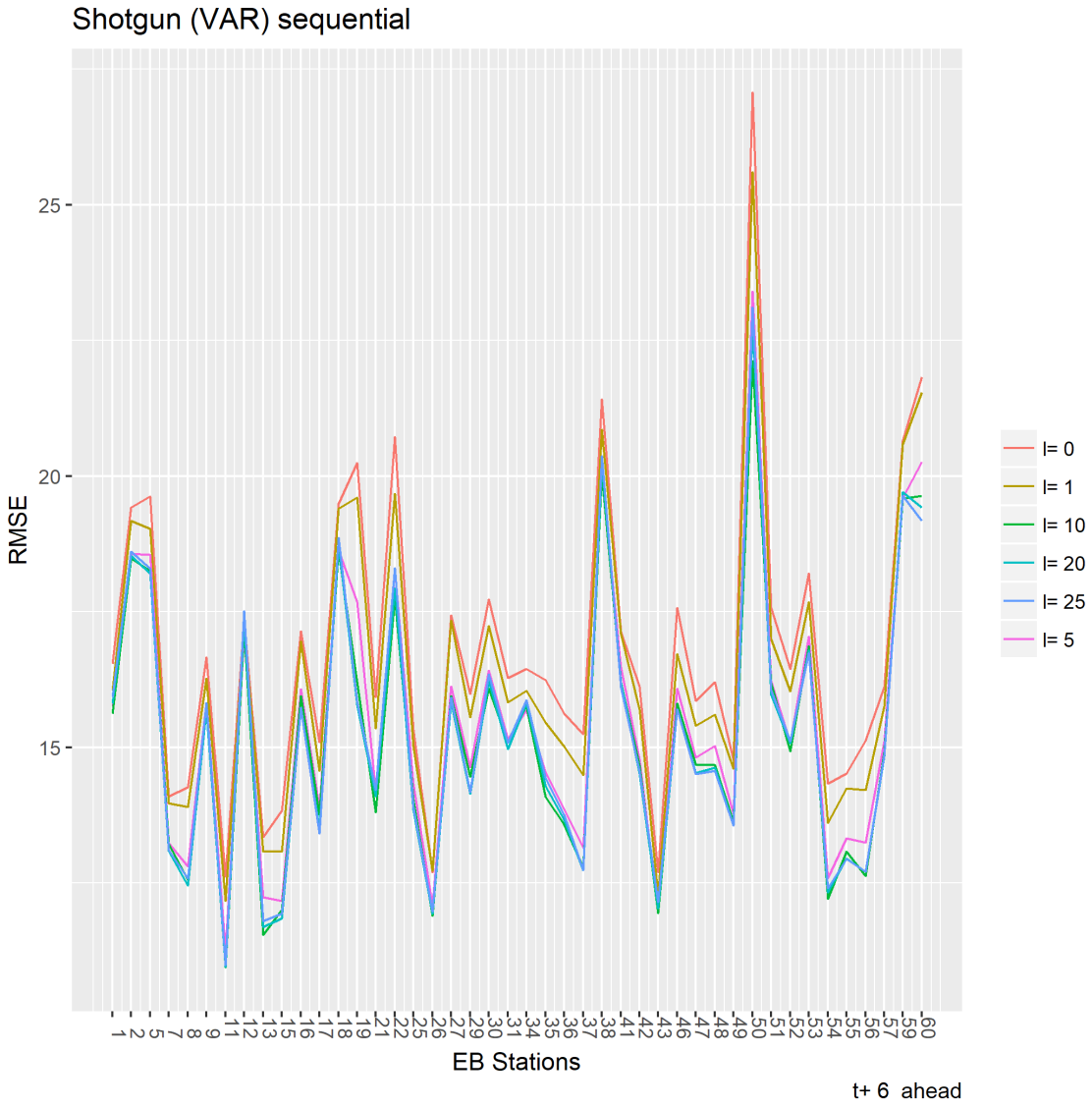


Figure 4.18: Shotgun (VAR) sequential models RMSE for $t + 6$

If we compare the models for step $t + 6$ figures 4.17 and 4.18 we observe a difference that was not observed in the sub-problem approach, the sequential method performs a little better than the parallel method, this particular case will be discussed in sections 5.1 and 5.2.

The Shotgun models'performances were discussed, lets now take a look to the GRock models. Figures 4.19, 4.20, 4.21 and 4.22 shows the error obtained by the GRock models. The figures 4.19 and 4.20 are representative of the models for $t + 1$ step and we can see these models does not present any gain, preforming way worst than the OLS model, and the scenario is worst as the forecast step goes further as we can see in figures 4.21 and

4.22 that represent the error of forecast step $t + 6$.

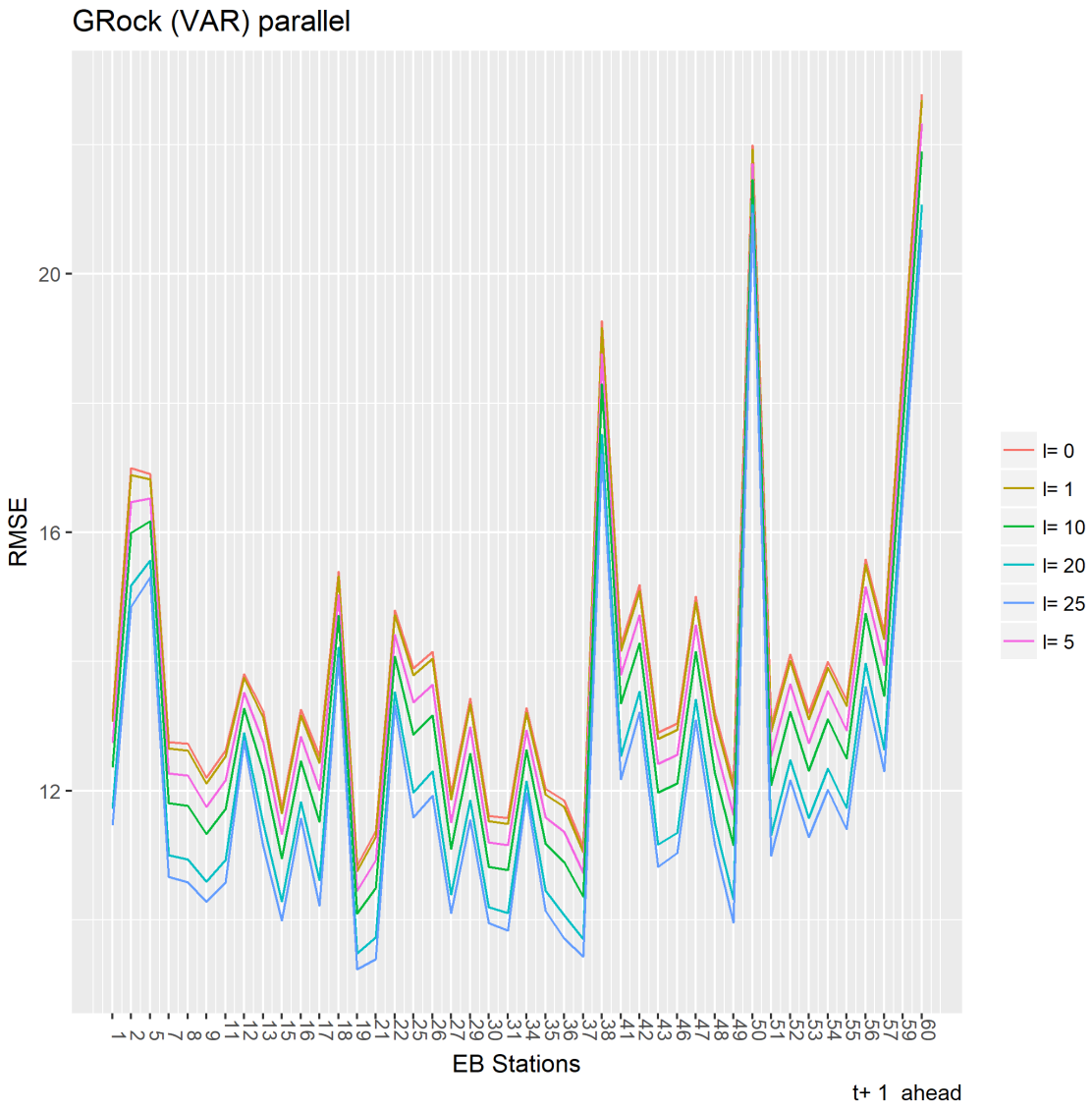


Figure 4.19: GRock parallel models RMSE for $t + 1$

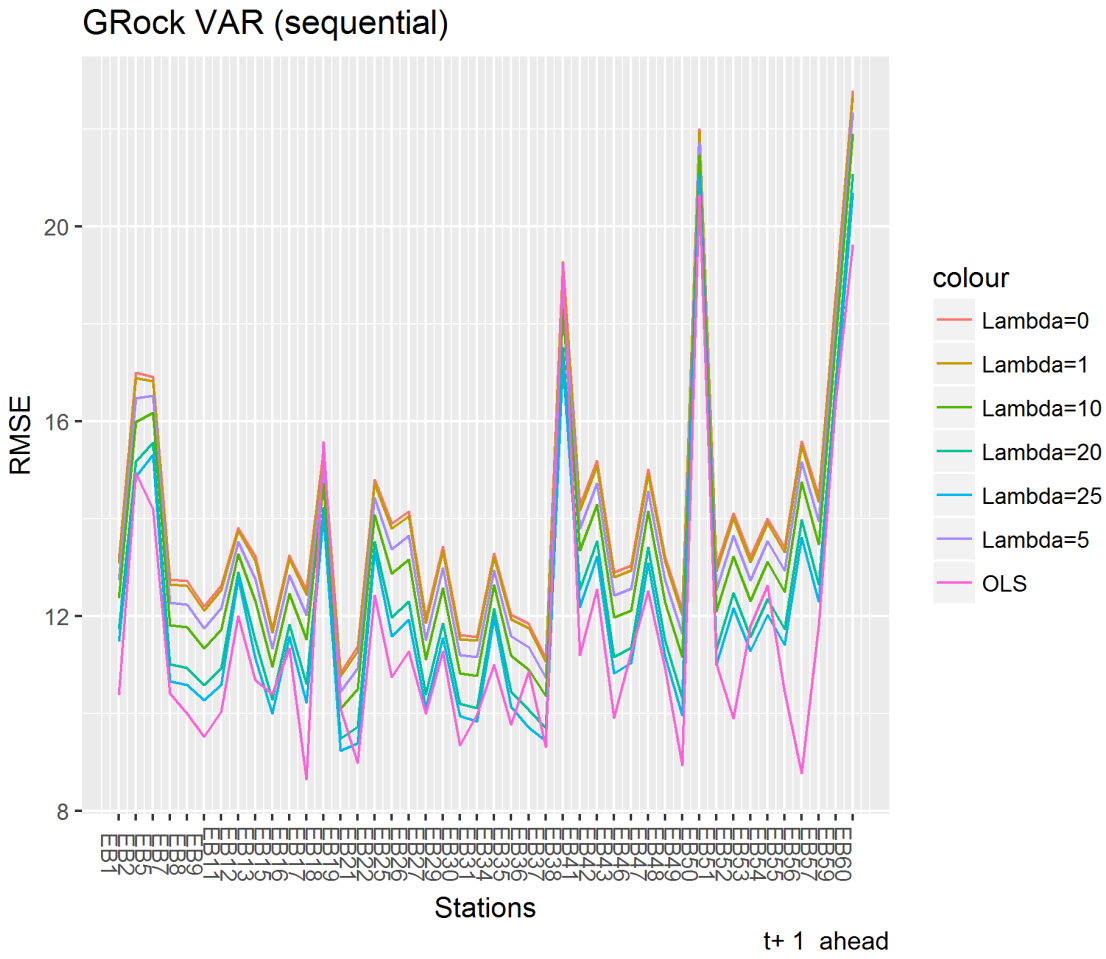


Figure 4.20: GRock sequential models RMSE for $t + 1$

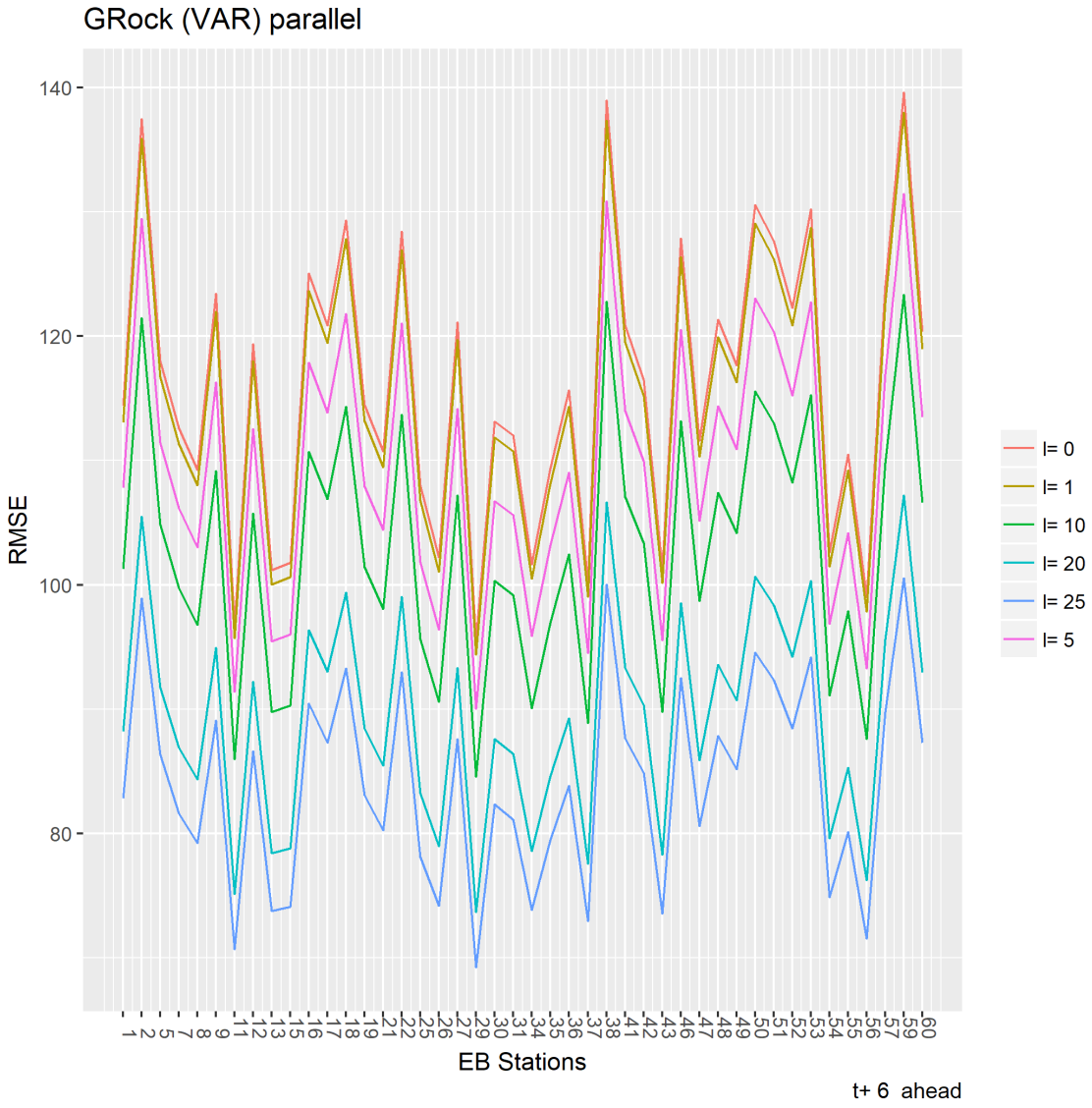


Figure 4.21: GRock parallel models RMSE for $t + 6$

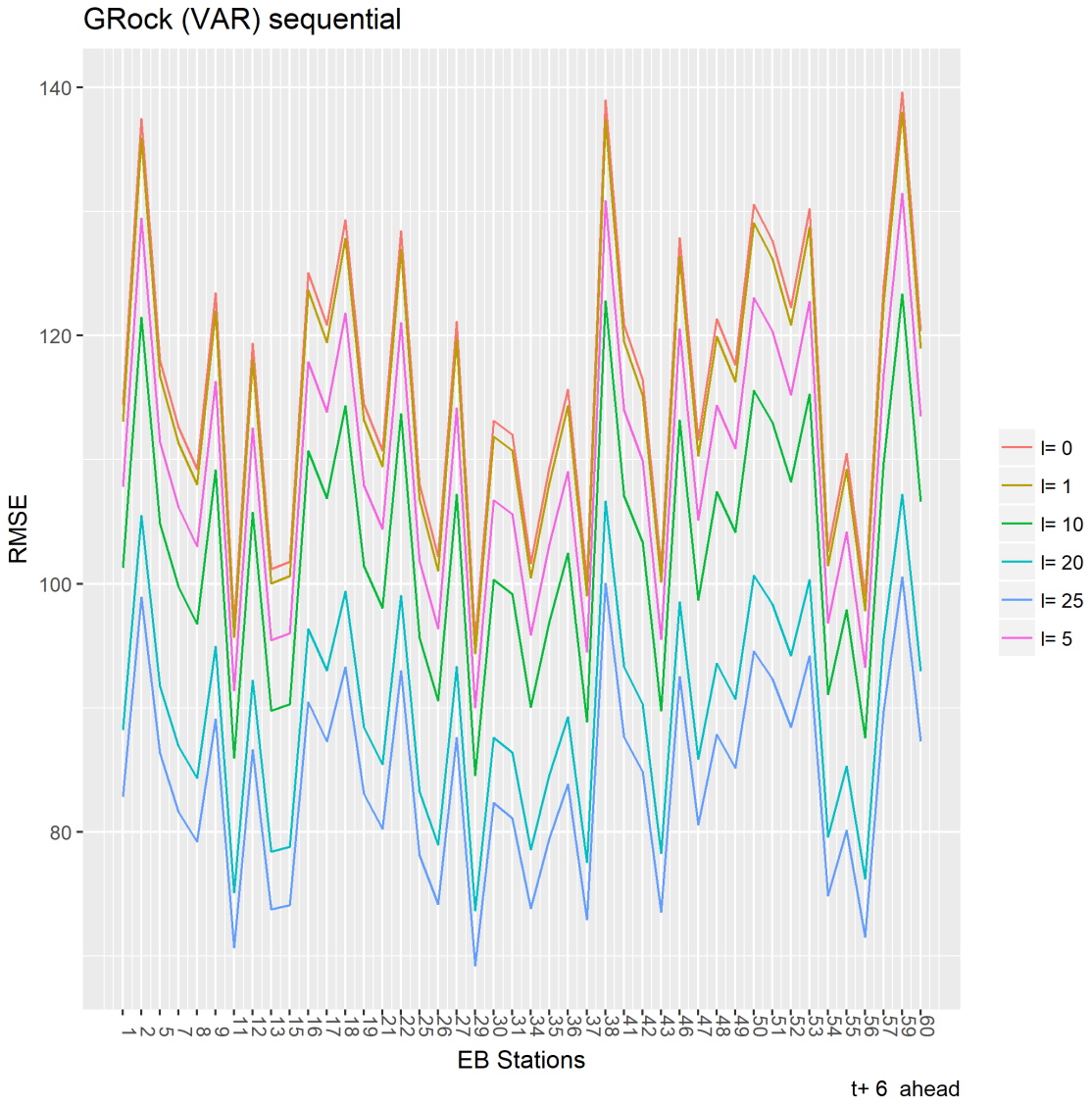


Figure 4.22: GRock sequential models RMSE for $t + 6$

4.3.2 Forecast

In this section, we will see some examples of the forecast to the train models of Shotgun and GRock train with parallel programming techniques. In figure 4.23 we can see that for step $t + 1$ the Shotgun models presents a forecast quite similar in value as well as in trajectory of the observed values. In figure 4.24 we can see that for step $t+6$, these models maintain the similarity in terms of trajectory, but the in terms of the value starts to deviate a little. Even so the forecast values these models are quite better than the ones from the

OLS model.

The previous observations made for shotgun models for the sub-problem approach is also observed for the VAR approach, meaning that these models fit better than the OLS model. figures 4.25 shows the fit of the models in the period 2012-06-28 until 2012-07-03 for step $t+1$ and figure 4.26 for the same period but for step $t+6$. We can see that in both cases the Shotgun models are quite good.

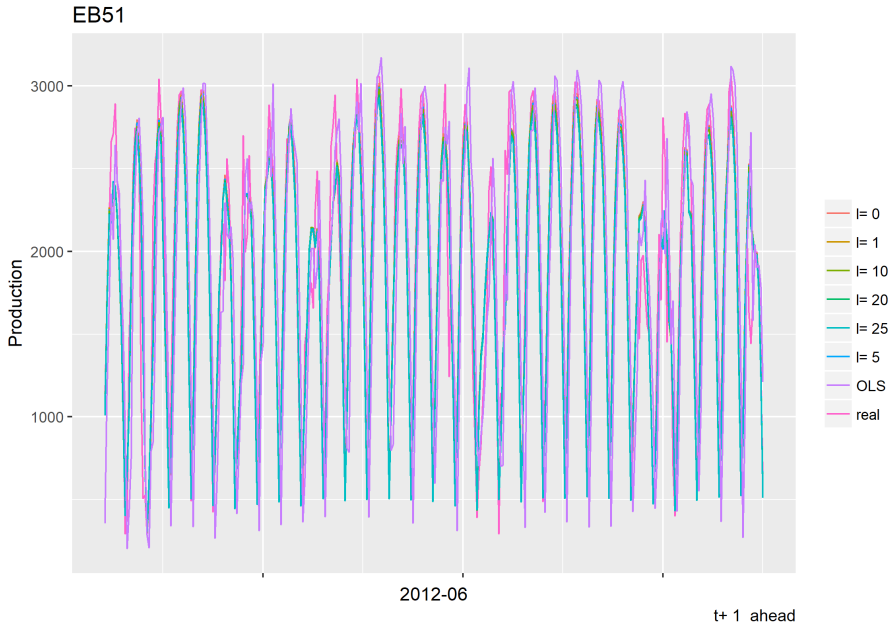


Figure 4.23: EB51 Shotgun (VAR) parallel models'step $t+1$ forecast for 2012-06

Let's now look at GRock models for the VAR approach using Parallel techniques.

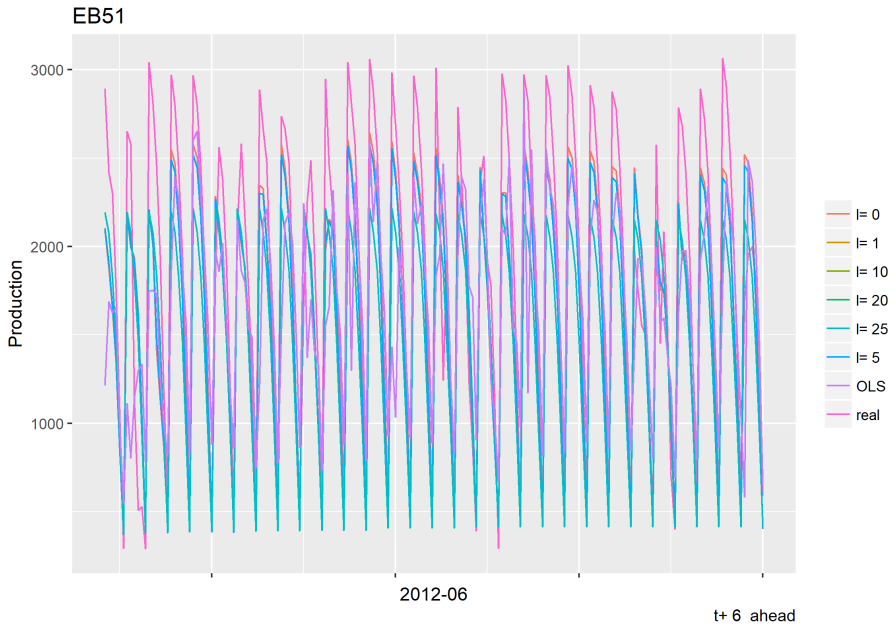


Figure 4.24: EB51 Shotgun (VAR) parallel models'step $t + 6$ forecast for 2012-06

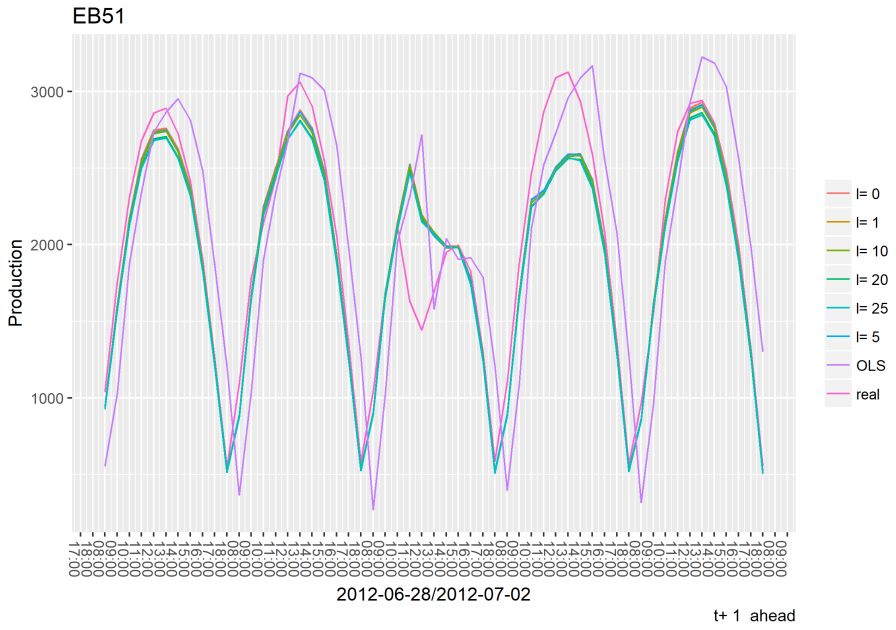


Figure 4.25: EB51 Shotgun (VAR) parallel models'step $t + 1$ forecast for period 2012-06-28/012-07-02

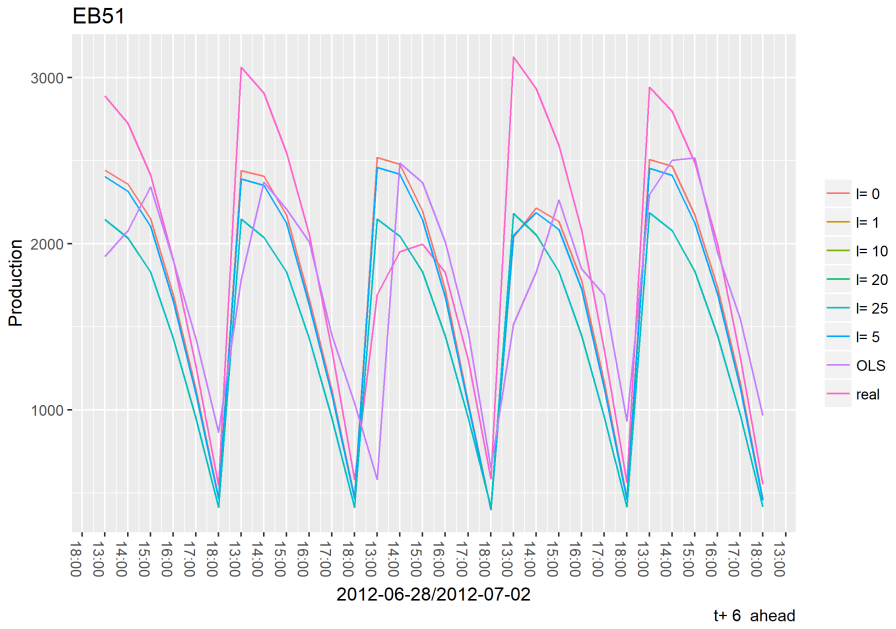


Figure 4.26: EB51 Shotgun (VAR) parallel models'step $t+6$ forecast for period 2012-06-28/012-07-02

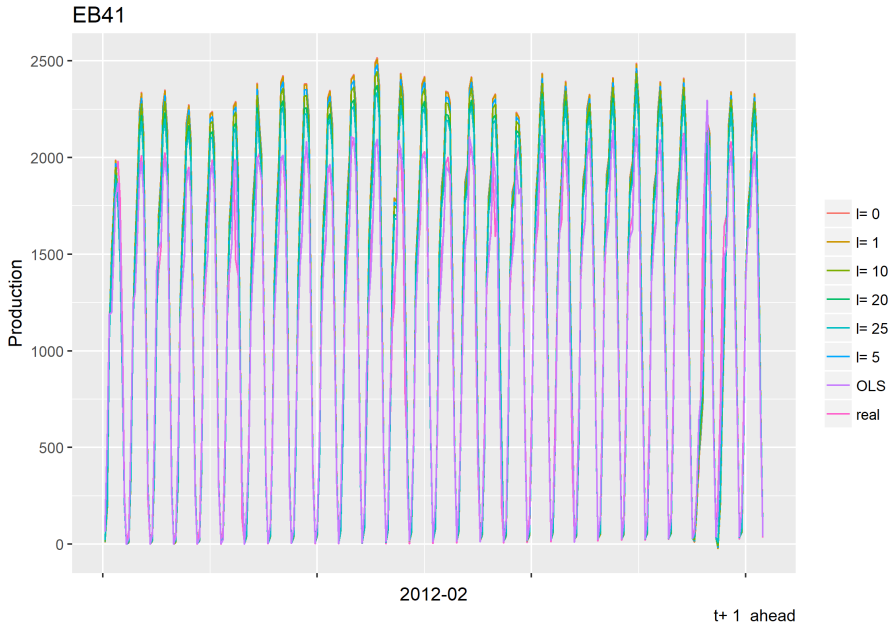


Figure 4.27: EB41 GRock (VAR) parallel models'step $t+1$ forecast for 2012-02

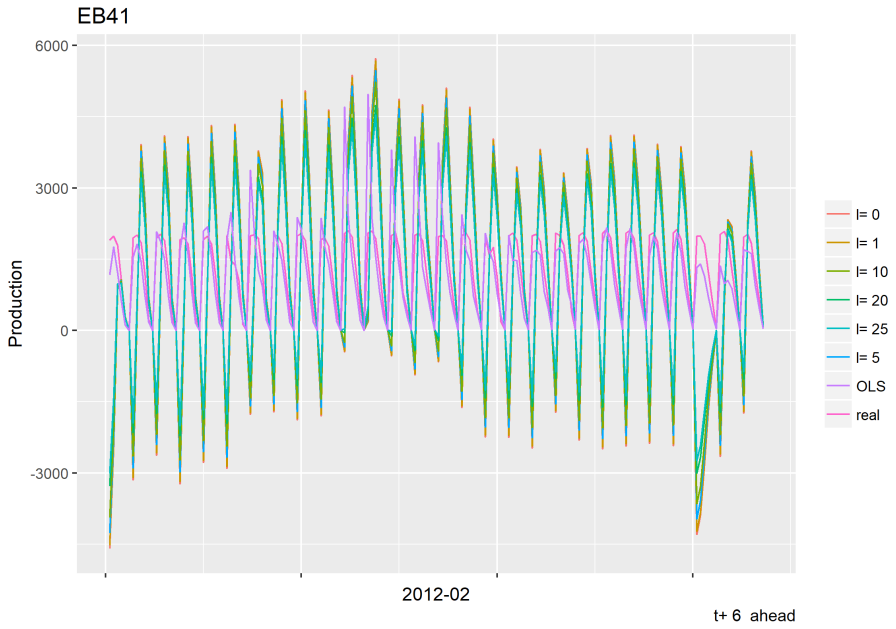


Figure 4.28: EB41 GRock (VAR) parallel models'step $t + 6$ forecast for 2012-02

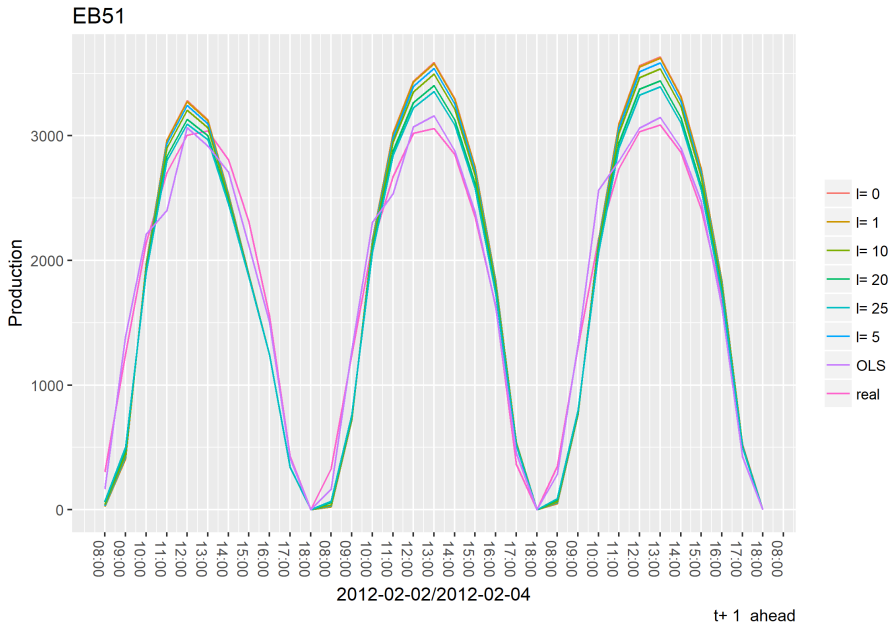


Figure 4.29: EB51 GRock (sub-problem) parallel models'step $t + 1$ forecast for period 2012-02-02/012-02-04

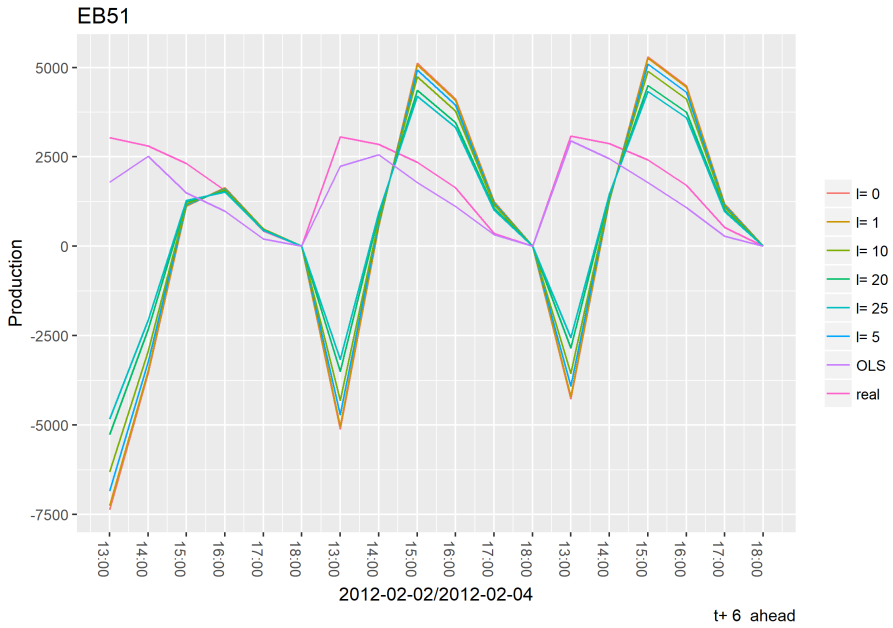


Figure 4.30: EB51 GRock (sub-problem) parallel models'step $t + 6$ forecast for period 2012-02-02/012-02-04

As we can see on figures 4.27,4.28,4.29 and 4.30, These models are different from the OLS model, when focus in the step $t + 1$, but when we look to a further step such as $t + 6$ we can see that these models performances is way worst.

Chapter 5

Conclusions and Future work

In this chapter, we will look for a global overview of the models and their results. In section 5.2 it will present some future works that we consider as the most relevant to introduce these models into a real environment, from the computation performance as well to the algorithm adaptation.

5.1 Conclusions

In terms of models, we could see that the Shotgun models performed way better in all of the scenarios.

GRock models in the problem approach as sub-problems also brought some gains. The GRock runtime to train the models is way lower than the Shotgun, which could be considered an alternative to the OLS model since he a slightly better model than the OLS. Even in the further steps forecast the GRock performs better than the OLS model.

When we board the shotgun models we need to take into consideration some situations that we observe in the train of these models. It is undeniable the performance of these models, they bring lots of gain in terms of performance principally in the further steps, that is a better way and even a more secure way to forecast the PV production within $6 - hours$ ahead. This in the OLS model was not visible, and the forecast for $6 - hours$ ahead came with a higher error. As observed the performance of the models is quite similar, but if we took in consideration tables 5.1 and 5.2 we can verify which models performed better in each situation. In those tables are represented the weight of each model, i.e. those tables represent the number (and respective percentage) of the minimum RMSE by λ in this way we can see which model performs better for the forecast of step $t + 1$.

As we can see on table 5.1 the better model for all the approaches fo the GRock is the model $\lambda = 25$. For the shotgun models, this is not so obvious we must consider a different model for each approach. We can see that for step $t + 1$ forecast model $\lambda = 0$ is the best

Table 5.1: GRock Models'RMSE for step $t + 1$

		λ=0		λ=1		λ=5		λ=10		λ=20		λ=25	
		#	%	#	%	#	%	#	%	#	%	#	%
Sub-Problems	Sequential	1	2,3							2	4,6	41	93,2
	Parallel									4	9,1	41	90,9
VAR	Sequential											44	100
	Parallel											44	100

model when the model trains in a sequential computer programming paradigm for the sub-problem approach and for the parallel computer programming paradigm for the VAR approach. For the remain approaches (parallel computer programming paradigm for the subproblem approach and sequential computer programming paradigm for the VAR) the model that has the best performance is the model $\lambda = 10$.

Table 5.2: Shotgun Models'RMSE for step $t + 1$

		λ=0		λ=1		λ=5		λ=10		λ=20		λ=25	
		#	%	#	%	#	%	#	%	#	%	#	%
Sub-Problems	Sequential	15	34,1	14	31,8	9	20,5	4	9,1	1	2,3	1	2,3
	Parallel	2	4,6	6	13,6	6	13,6	23	52,3	5	11,4	2	4,6
VAR	Sequential	4	9,1	4	9,1	3	6,8	22	50,0	1	2,3	10	22,7
	Parallel	19	43,2	12	27,3	7	15,9	3	6,8	2	4,6	1	2,3

The values from table 5.2 are a resume from the obtained RMSE values that can be consulted on appendix A for a deeper analysis on the obtained RMSE for Shotgun models in the forecast to step $t + 1$.

To a clear vision of the aforementioned we can take a look into the RMSE graphs for steps $t + 1$ and $t + 6$ for the top 3 models for each approach to Shotgun. If we take a look in figures 5.1, 5.2, 5.3, 5.4, B.5, B.6, B.7 and B.8 we can see the error behaviour for steps $t + 1$ and $t + 6$ forecasts and selected the model that better performs in both situations

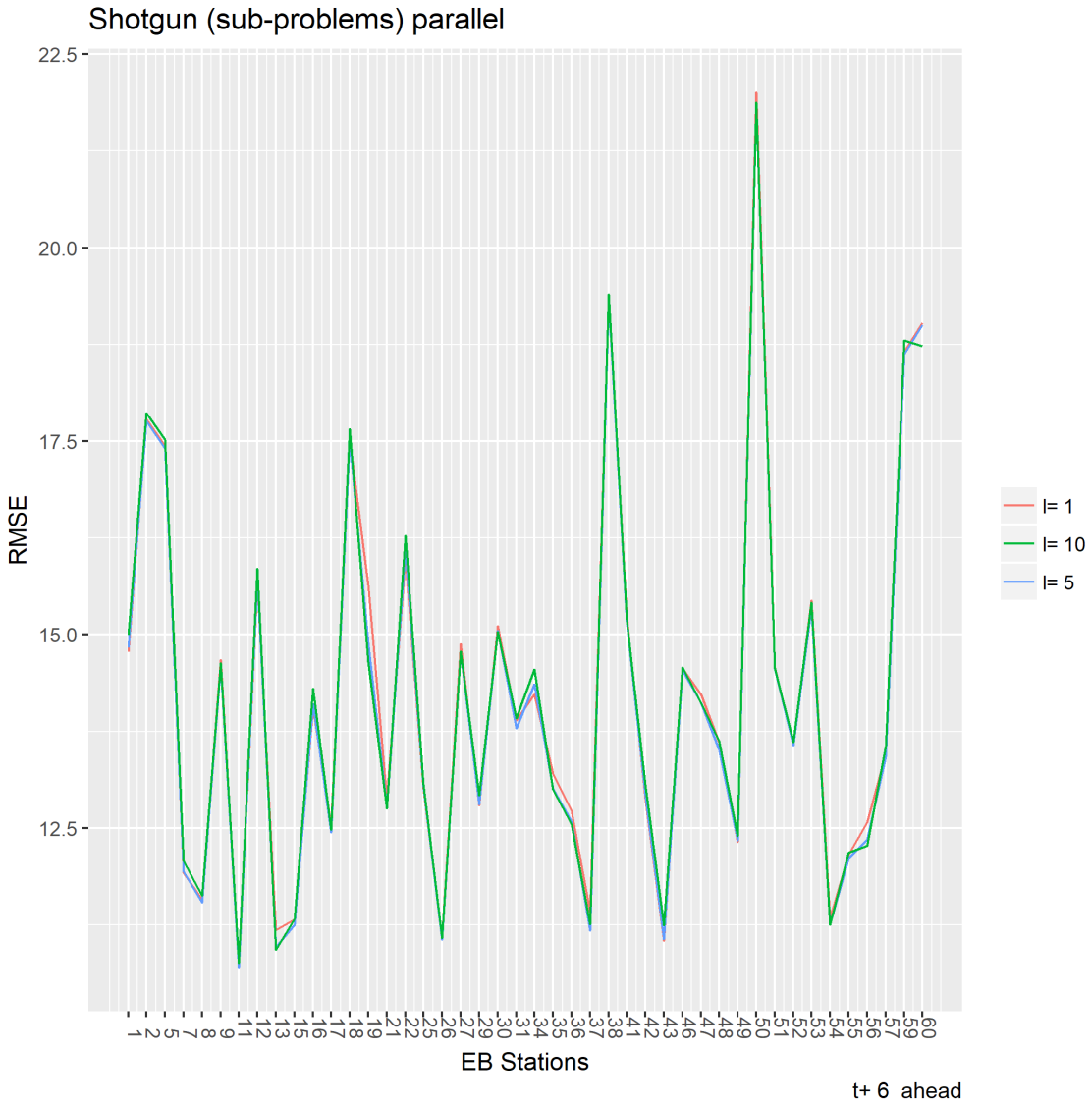


Figure 5.1: Shotgun (Sub-problems) parallel RMSE for step $t + 6$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 10$

We can see on figure 5.1 model $\lambda = 10$ is the best model for step t+6, this model is also the better model for step t+1, as we can see on figure B.7, this should be the selected model for the sub-problem approach trained with parallel computer programming paradigm.

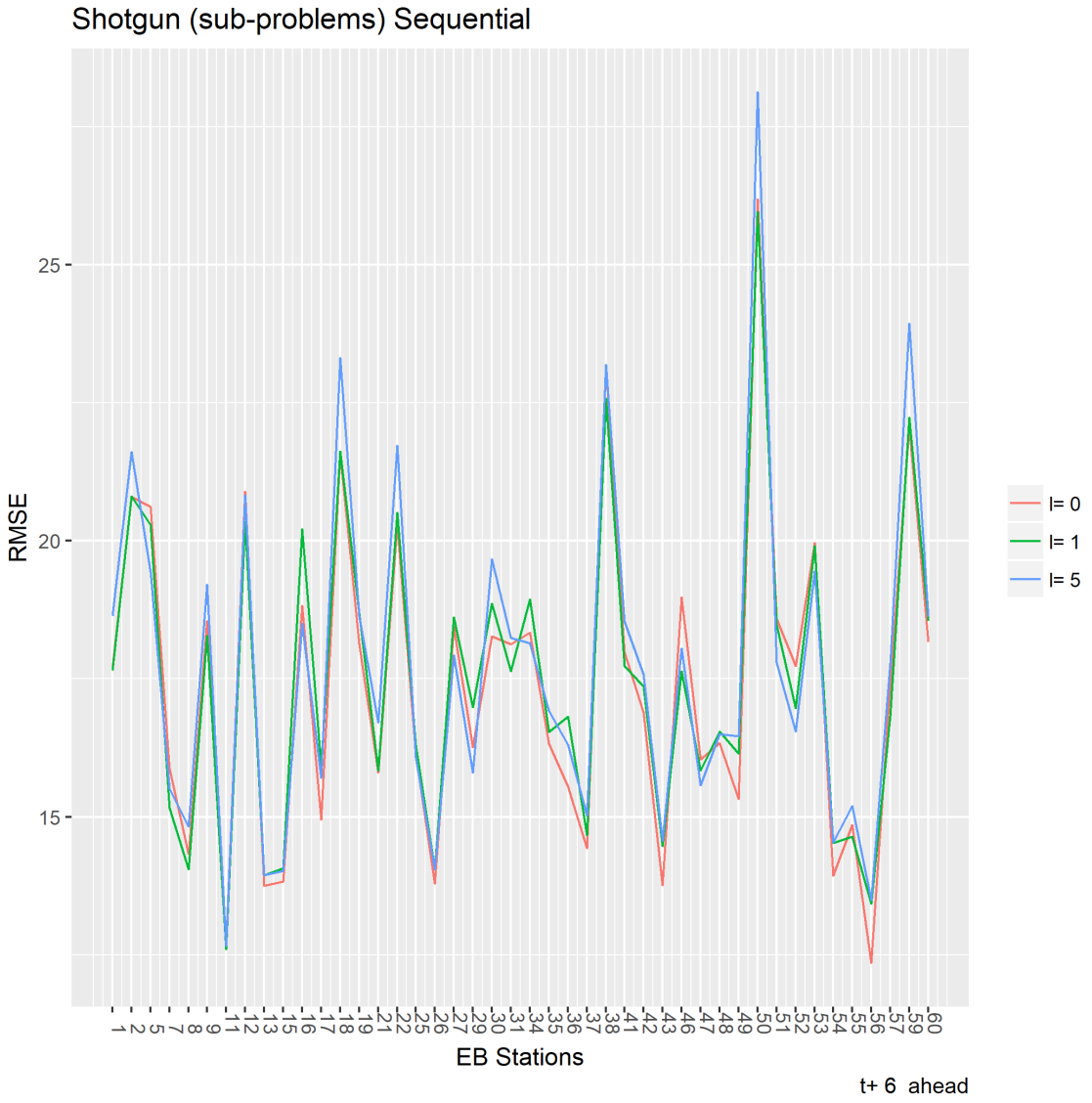


Figure 5.2: Shotgun (sub-problem) sequential RMSE for step $t+6$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$

We can see on figure 5.2 model $\lambda = 0$ is the best model for step $t+6$, this model is also the better model for step $t+1$, as we can see on figure B.5, this should be the selected model for the sub-problem approach trained with sequential computer programming paradigm.

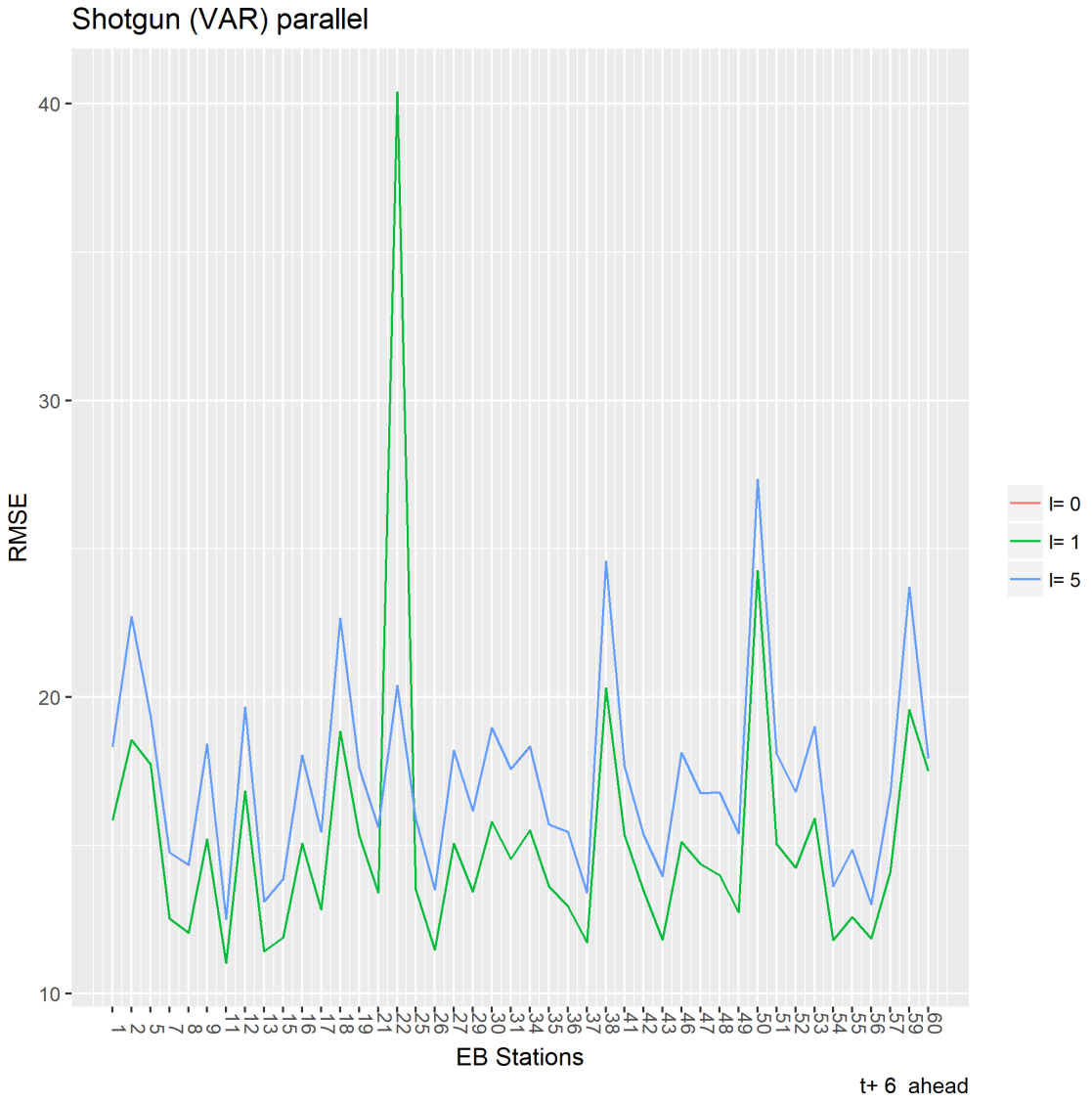


Figure 5.3: Shotgun (VAR) parallel RMSE for step $t + 6$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$

If we look for 5.3 we can see that the model $\lambda = 1$ outperforms the remain models for the VAR approach when trained with the parallel computer programming paradigm. Even for step $t + 1$ the better model is $\lambda = 0$, as all the models RMSE are very close to each other for steps $t + 1$, as we can see in figure B.8 it will be interesting consider the model $\lambda = 1$ to be the better one, since it will bring a greater gain in a further step forecasts.

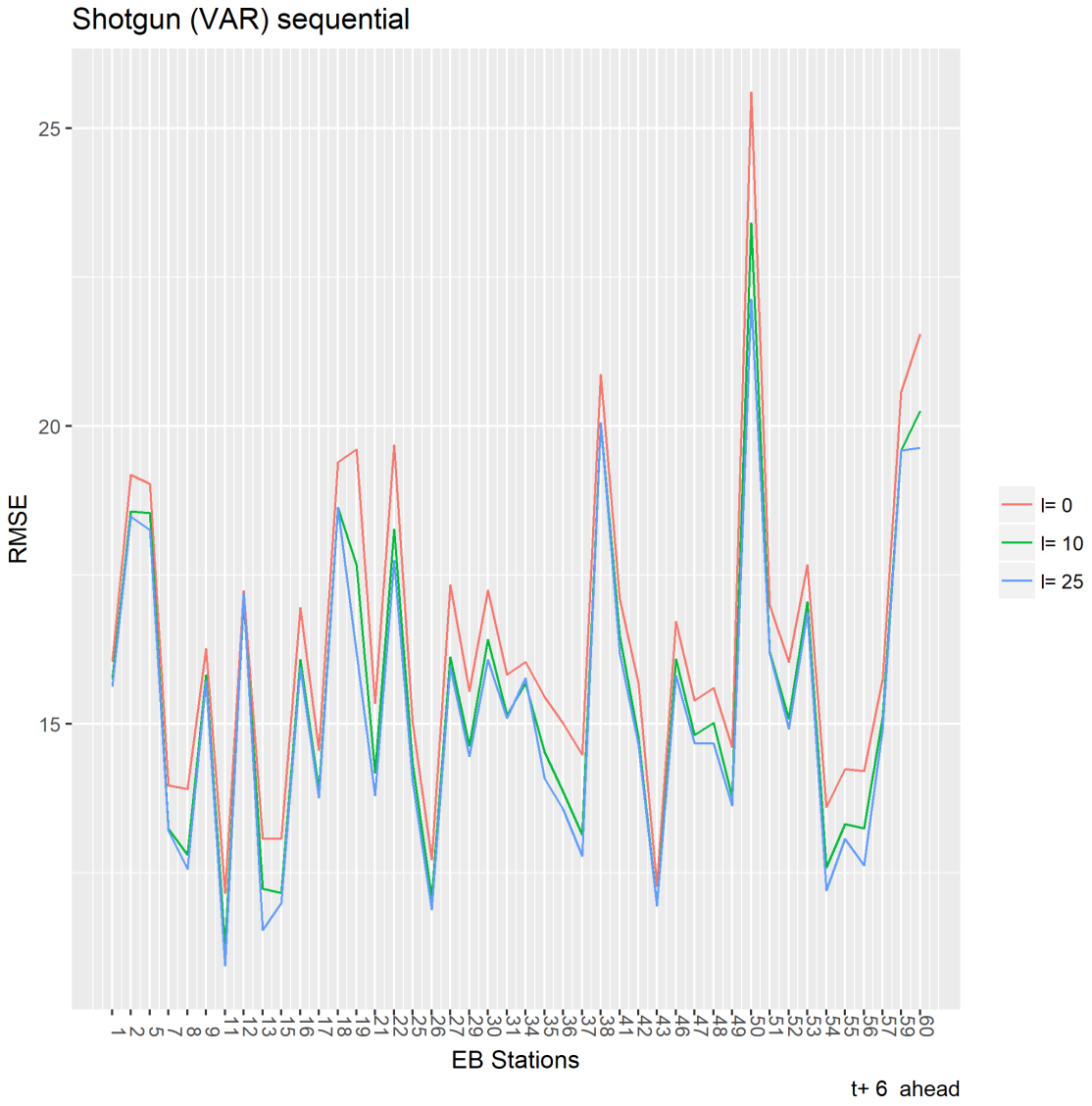


Figure 5.4: Shotgun (VAR) sequential RMSE for step $t+6$. Models $\lambda = 0, \lambda = 10$ *and* $\lambda = 25$

The same principle can be applied to models for VAR approach when trained with sequential computer programming paradigm, in this case, we can consider the better model as to be model $\lambda = 25$. We can see in B.6 that the RMSE values are very similar and in figure 5.4 we can see that the model that has the better performance is $\lambda = 25$.

Although the shotgun models presented a better performance, this algorithm struggles to converge. When train the model this algorithm could not converge, the stop condition was the *Max Iterations* condition, what reveals a runtime way too expensive, the higher *Max Iterations* the better was the performance of the models. This fact leads to the

difference that we could observe in figures 4.17 and 4.18 beside the train model of the sequential technique has a higher *Max Iterations* than value used in the Parallel technique, in the parallel technique the *Max Iterations* value $\approx \frac{1}{3}$ from the value in the sequential technique and the result in the parallel techniques was also very good and similar to the sequential technique.

Even with this expensive train runtime, these models should be taken into consideration to the forecast of the PV in a SMART Grid since the performance of these models are considerably better from the ones obtained using the OLS model.

5.2 Future Work

This work leads to a further learn into the LASSO-VAR problem, where the PV production forecast is more accurate and it can be performed for the entire SMART Grid.

In terms of future work we can approach this in two different angles, according to the algorithm

5.2.1 Shotgun

For this algorithm, the future work is the study and perfect the convergence of algorithm, trying a faster convergence of the algorithm. Another subject that can be addressed is the study of the update step-size effect on the model and its performance. This algorithm according to the Bradley et al. (2011) is very dependent of this step-size.

5.2.2 GRock

This algorithm as it was presented in Peng et al. (2013) shows a better performance than the OLS model. These models converge quite fast. The future work related to this algorithms are related to his adaptation to the VAR problem, that as saw the performance of these models were very poor. If this algorithm could perform at the level of the Shotgun models, with this convergence rate, this algorithm will be a real contestant in the forecast PV production within a SMART Grid.

5.2.3 General

The general future work, besides those aforementioned, is the study of some of the other algorithms, and perfect the parallel approach. The results that were obtained, these techniques are relevant and deserves to be considered and further studied. Another approach can be the study of other algorithms that are presented in the section 2.5 of chapter 2 that were not considered in this work. Some of those algorithms brought some new approaches to the coordinate selections among other features.

Appendix A

RMSE Values

In this appendix we can see the RMSE values for the models Shotgun for the sequential and parallel. These models are for both problem approach (Sub-problems and VAR).

Table A.1: RMSE for Shotgun (sub-problem) parallel models for step $t + 1$

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB1	10.012604	10.001901	9.961691	9.905607	9.954736	10.187185
EB2	11.856840	11.697287	11.493662	11.542477	11.468698	11.594358
EB5	13.237737	13.319459	13.311989	13.292354	13.403681	13.477888
EB7	8.287733	8.343336	8.327494	8.282804	8.322033	8.352220
EB8	8.176895	8.111272	8.054746	7.960440	7.978208	7.988929
EB9	8.537172	8.532351	8.541390	8.541626	8.567859	8.787593
EB11	8.448360	8.369820	8.344315	8.301051	8.444208	8.550338
EB12	12.054719	12.021554	11.992241	11.993677	12.040306	12.117337
EB13	9.115038	9.065447	9.008761	8.962394	9.083433	8.984940
EB15	7.917593	7.894373	7.903507	7.862390	7.918938	8.003697
EB16	9.741968	9.689950	9.709809	9.712051	9.800741	9.831745
EB17	7.168527	7.139815	7.196529	7.107344	7.110423	7.130272
EB18	11.585297	11.526667	11.509256	11.527897	11.537014	11.551405
EB19	8.370002	8.411302	8.418606	8.471749	8.665525	8.940570
EB21	7.999457	7.978554	7.974965	7.968671	8.086817	8.117622

Table A.2: RMSE for Shotgun (sub-problem) parallel models for step t + 1- Cont.

	λ= 0	λ= 1	λ= 5	λ= 10	λ= 20	λ= 25
EB22	11.362083	11.303149	11.368330	11.403417	11.502820	11.581592
EB25	8.934655	8.960372	8.839132	8.740756	8.763349	8.747383
EB26	9.117829	9.073920	9.026726	8.942623	9.132385	9.116956
EB27	8.541019	8.532000	8.510854	8.494571	8.494279	8.509081
EB29	9.386635	9.408061	9.312703	9.345482	9.351901	9.491343
EB30	8.819528	8.804019	8.770637	8.746348	8.781866	8.858063
EB31	8.607854	8.622991	8.548083	8.472371	8.477261	8.500738
EB34	10.825654	10.806290	10.749485	10.786860	10.853040	10.914581
EB35	8.953047	8.930773	8.928984	8.894063	8.970871	9.036797
EB36	7.805876	7.794643	7.772637	7.762326	7.971297	7.857420
EB37	8.276581	8.259640	8.301438	8.270397	8.397655	8.481372
EB38	13.607221	13.505489	13.326335	13.294958	13.293013	13.216999
EB41	10.056249	10.027225	10.018162	9.965688	9.955859	9.967938
EB42	10.469613	10.428965	10.421173	10.371709	10.444458	10.469130
EB43	8.502856	8.428791	8.396254	8.308236	8.372871	8.415767
EB46	9.034764	9.029133	8.956635	8.912183	8.985829	8.969996
EB47	10.966339	10.973485	10.935756	10.953522	11.080135	11.126813
EB48	8.943622	8.976341	8.905118	8.826114	8.824188	8.863662
EB49	7.750659	7.738478	7.683253	7.642383	7.673060	7.710376
EB50	17.134981	17.063227	17.107324	17.106509	17.680316	18.461746
EB51	9.074949	9.060283	9.006212	8.947864	8.974466	9.004241
EB52	10.209596	10.203685	10.194681	10.173920	10.202197	10.246520
EB53	9.509021	9.509725	9.427399	9.376721	9.366019	9.521764
EB54	9.463582	9.454234	9.437897	9.449006	9.646629	9.750169
EB55	8.816317	8.842996	8.778258	8.721913	8.776061	8.812682
EB56	10.444871	10.420157	10.491699	10.524862	11.021146	10.704504
EB57	9.734555	9.727283	9.596719	9.587324	9.624886	9.894218
EB59	12.998403	13.054870	12.926903	12.918775	12.975630	12.940753
EB60	17.985449	17.944283	17.797992	17.858418	17.703980	17.688658

Table A.3: RMSE for Shotgun (sub-problem) sequential models for step $t + 1$

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB1	11.209796	11.126143	11.438507	11.402958	11.533977	11.722914
EB2	14.188032	14.186172	14.159321	14.194267	14.259599	14.217296
EB5	14.545750	14.525662	14.509212	14.485631	14.632107	14.587994
EB7	9.777716	9.722734	9.739983	9.874549	9.880582	10.044757
EB8	9.477842	9.505685	9.574551	9.532564	9.677647	9.672402
EB9	9.439770	9.605614	9.623762	9.479490	9.641332	9.768020
EB11	9.571469	9.497264	9.742453	9.888666	9.784760	9.858355
EB12	12.938299	12.909846	13.078662	13.344381	13.379781	13.357155
EB13	10.040396	10.044234	9.996581	10.076302	10.424335	10.245873
EB15	9.157335	9.122967	9.173295	9.207520	9.239352	9.392018
EB16	11.155310	11.274065	11.132725	11.188993	11.528043	11.445249
EB17	8.925687	8.950253	8.923940	8.921268	8.971884	9.020993
EB18	14.068169	13.942978	14.051687	13.974361	14.121080	14.184823
EB19	8.995577	9.117527	9.259732	9.265670	9.346293	9.918819
EB21	8.605507	8.775161	8.635598	8.661860	8.948407	9.100026
EB22	13.105897	13.283118	13.258681	13.305895	13.431611	13.797413
EB25	10.399565	10.385154	10.382268	10.490270	10.465437	10.628269
EB26	10.610074	10.629937	10.586941	10.620855	10.814326	10.920255
EB27	9.458662	9.406884	9.402302	9.406905	9.440280	9.512757
EB29	10.782565	10.751947	10.795546	10.838303	11.061018	11.016545
EB30	9.631920	9.678021	9.983171	9.789557	9.927088	9.924309
EB31	9.237423	9.325305	9.407457	9.385973	9.409450	9.471127
EB34	11.925439	11.939356	12.087304	12.229619	12.216747	12.372616
EB35	9.583744	9.511819	9.535535	9.653072	9.982534	10.046288
EB36	8.554162	8.622089	8.539236	8.756847	8.759366	8.726709
EB37	9.164184	9.024016	9.145765	9.373711	9.492376	9.430650
EB38	16.873326	16.710429	16.797467	16.704690	16.606570	16.673399
EB41	10.669048	10.659155	10.747342	10.685175	10.667945	10.688922
EB42	12.350260	12.387874	12.404078	12.410345	12.514839	12.629717
EB43	9.769172	9.785982	9.806652	9.886820	9.960225	9.983347
EB46	10.400042	10.401158	10.352586	10.399345	10.437833	10.561319
EB47	12.027143	11.948754	11.988415	12.061073	12.191662	12.179869
EB48	10.365404	10.399178	10.304222	10.301324	10.455377	10.496277
EB49	8.903079	9.054338	8.915210	8.927074	9.115909	9.022421
EB50	21.003708	20.943762	21.147959	21.232847	21.422952	21.432801
EB51	10.232475	10.199870	10.331493	10.286389	10.401396	10.455010

Table A.4: RMSE for Shotgun (sub-problem) sequential models for step $t + 1$ - Cont

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB52	11.382310	11.424024	11.431616	11.695268	11.525491	11.861363
EB53	10.602034	10.729765	10.682792	10.677022	10.880707	10.714481
EB54	10.988715	11.110116	11.191777	11.253204	11.346221	11.575237
EB55	10.560154	10.575604	10.586435	10.832034	10.680567	10.761148
EB56	11.495008	11.505254	11.505374	11.478040	11.546062	11.479667
EB57	11.460813	11.444688	11.590185	11.610365	11.764324	11.719478
EB59	15.877829	15.785741	15.756735	15.786396	15.883269	15.926475
EB60	18.746627	18.630899	18.620637	18.592918	18.650082	18.123265

Table A.5: RMSE for Shotgun (VAR) parallel models for step $t + 1$

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB1	10.665952	10.625750	10.696610	10.807577	10.943455	11.011334
EB2	12.724583	12.651605	12.710216	12.632068	12.846314	12.872890
EB5	14.119006	14.058746	14.085840	14.121045	14.083237	14.178319
EB7	9.103968	9.099692	9.161370	9.236481	9.299514	9.446775
EB8	8.780395	8.815582	8.774133	8.843280	8.899403	8.993186
EB9	9.239470	9.216833	9.195556	9.266859	9.359851	9.389846
EB11	9.082354	9.097328	9.160515	9.184126	9.315403	9.434198
EB12	12.673777	12.669160	12.694622	12.743400	12.926740	13.028757
EB13	9.621189	9.749205	9.714837	9.688322	9.849658	9.901933
EB15	8.628131	8.553155	8.658929	8.690436	8.744131	8.827409
EB16	10.572896	10.541531	10.588210	10.526270	10.730840	10.748156
EB17	7.941648	8.065549	8.029899	8.017641	8.043260	8.126323
EB18	12.713835	12.772820	12.745986	12.791497	12.859733	13.053347
EB19	9.073121	9.055801	9.123917	9.194723	9.409179	9.555423
EB21	8.499761	8.503213	8.521933	8.618303	8.802783	8.902010
EB22	13.215413	13.168717	12.985553	12.774765	12.721086	12.752822
EB25	9.628441	9.602343	9.697733	9.703976	9.821260	9.868167
EB26	9.778470	9.792120	9.823294	9.791563	9.937862	9.966852

Table A.6: RMSE for Shotgun (VAR) parallel models for step $t + 1$ - cont

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB27	8.959297	8.923850	8.993795	8.984736	9.037680	9.083018
EB29	10.100934	10.182273	10.129151	10.207267	10.245340	10.343201
EB30	9.429308	9.465960	9.506073	9.582071	9.720019	9.698616
EB31	8.991319	9.054918	8.987366	9.077259	9.104568	9.159902
EB34	11.463263	11.524909	11.628098	11.597604	11.738466	11.847535
EB35	9.509872	9.449744	9.542949	9.583424	9.723889	9.809498
EB36	8.303484	8.327932	8.325977	8.340030	8.445074	8.488888
EB37	8.932369	8.961635	8.996877	9.049976	9.160925	9.351486
EB38	14.620485	14.633555	14.561646	14.648728	14.626025	14.662234
EB41	10.472043	10.484694	10.452113	10.491764	10.533750	10.507888
EB42	11.411532	11.496782	11.413600	11.540042	11.546437	11.661908
EB43	9.113517	9.134220	9.124291	9.224713	9.318149	9.390601
EB46	9.741903	9.791242	9.877178	9.854896	9.971411	9.972026
EB47	11.626979	11.632774	11.662690	11.669133	11.777244	11.827230
EB48	9.664486	9.698822	9.712971	9.741334	9.815352	9.870601
EB49	8.391398	8.343886	8.389477	8.495626	8.563236	8.574860
EB50	19.370856	19.510403	19.784692	19.590385	19.769616	19.791799
EB51	9.812268	9.730085	9.721520	9.871570	9.940913	9.930556
EB52	10.924502	10.868753	10.919280	11.043302	11.077608	11.126265
EB53	10.163562	10.198336	10.190414	10.231500	10.289093	10.369056
EB54	10.426904	10.354985	10.484538	10.535886	10.665387	10.802305
EB55	9.696142	9.755260	9.821364	9.796089	9.969708	10.038167
EB56	11.240105	11.243957	11.219188	11.244791	11.218100	11.248188
EB57	10.672590	10.649964	10.635292	10.741745	10.802882	10.954785
EB59	14.350273	14.337955	14.345249	14.312973	14.398551	14.415119
EB60	18.468650	18.551048	18.226164	18.276358	18.138643	18.008690

Table A.7: RMSE for Shotgun (VAR) sequential models for step $t + 1$

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB1	10.027665	10.041143	9.907361	9.801892	9.907258	9.951887
EB2	12.384064	12.011183	11.655190	11.557796	11.542091	11.329968
EB5	12.717599	12.764940	12.759193	12.917844	13.088269	13.034507
EB7	8.413031	8.361496	8.297907	8.268324	8.311341	8.349406
EB8	8.303415	8.227405	8.093144	7.916366	7.942831	7.914848
EB9	8.234984	8.213827	8.258266	8.301793	8.376140	8.373069
EB11	8.455508	8.486287	8.384233	8.110261	8.187524	8.198752
EB12	11.454541	11.522492	11.602829	11.590692	11.917472	11.850973

Table A.8: RMSE for Shotgun (VAR) sequential models for step $t + 1$ - Cont

	λ=0	λ=1	λ=5	λ=10	λ=20	λ=25
EB13	9.262514	9.197713	8.878939	8.816613	8.814769	8.767165
EB15	8.065865	8.114252	7.953734	7.836732	7.887419	7.862377
EB16	9.704685	9.599857	9.427050	9.406739	9.505546	9.509283
EB17	7.042478	7.034151	6.992339	6.955619	7.091199	6.964848
EB18	11.800862	11.542340	11.314088	11.176373	11.379389	11.279215
EB19	8.188251	8.202541	8.065439	8.057486	8.293380	8.335742
EB21	7.816370	7.790034	7.825210	7.849603	7.950338	7.983583
EB22	11.784932	11.670028	11.437068	11.239846	11.227034	11.163871
EB25	9.051958	8.978185	8.841523	8.587392	8.634360	8.612475
EB26	9.588619	9.593732	9.368735	9.181736	9.149234	9.112793
EB27	8.482971	8.467663	8.458671	8.452309	8.499862	8.490150
EB29	9.574105	9.497455	9.462161	9.243663	9.190317	9.157460
EB30	8.633334	8.561307	8.511582	8.484008	8.670954	8.711327
EB31	8.463992	8.587037	8.355974	8.297258	8.347291	8.382909
EB34	10.804513	10.523001	10.305122	10.440943	10.684514	10.528163
EB35	8.562945	8.493027	8.618893	8.601917	8.756966	8.751877
EB36	7.785218	7.706048	7.660264	7.683854	7.776784	7.775625
EB37	7.987037	8.026290	7.958808	7.922865	8.150582	8.127994
EB38	15.548081	15.054927	14.676714	13.860060	13.756526	13.761961
EB41	9.857355	9.817029	9.830477	9.747978	9.842589	9.833332
EB42	10.550017	10.585495	10.247274	10.188395	10.359411	10.314479
EB43	8.473145	8.470030	8.460279	8.229060	8.316450	8.344913
EB46	8.832381	8.940285	8.874376	8.740800	8.892187	8.892916
EB47	10.817422	10.838414	10.642048	10.653647	10.935768	10.893387
EB48	9.092363	9.020881	8.899929	8.760768	8.769919	8.749062
EB49	7.718090	7.701664	7.661776	7.565428	7.652677	7.671711
EB50	17.861384	17.692084	17.638500	16.666856	16.739866	16.307117
EB51	9.056430	8.986727	8.973245	8.864300	8.916608	8.895052
EB52	9.730497	9.788165	9.824985	9.791974	10.002739	9.976032
EB53	9.277028	9.379934	9.354667	9.216073	9.232325	9.255109
EB54	9.563529	9.511069	9.331924	9.323041	9.430756	9.451937
EB55	9.302087	9.176510	8.989231	8.907918	8.793240	8.757481
EB56	9.541285	9.571548	9.744272	9.596240	10.167892	10.005188
EB57	9.878972	9.720112	9.598006	9.546560	9.560144	9.557702
EB59	13.875663	13.834108	13.445643	13.064717	13.133901	12.985007
EB60	17.574259	17.248951	17.496836	17.329350	17.443331	17.585975

Appendix B

RMSE

This appendix intends to show the error plots of the models.

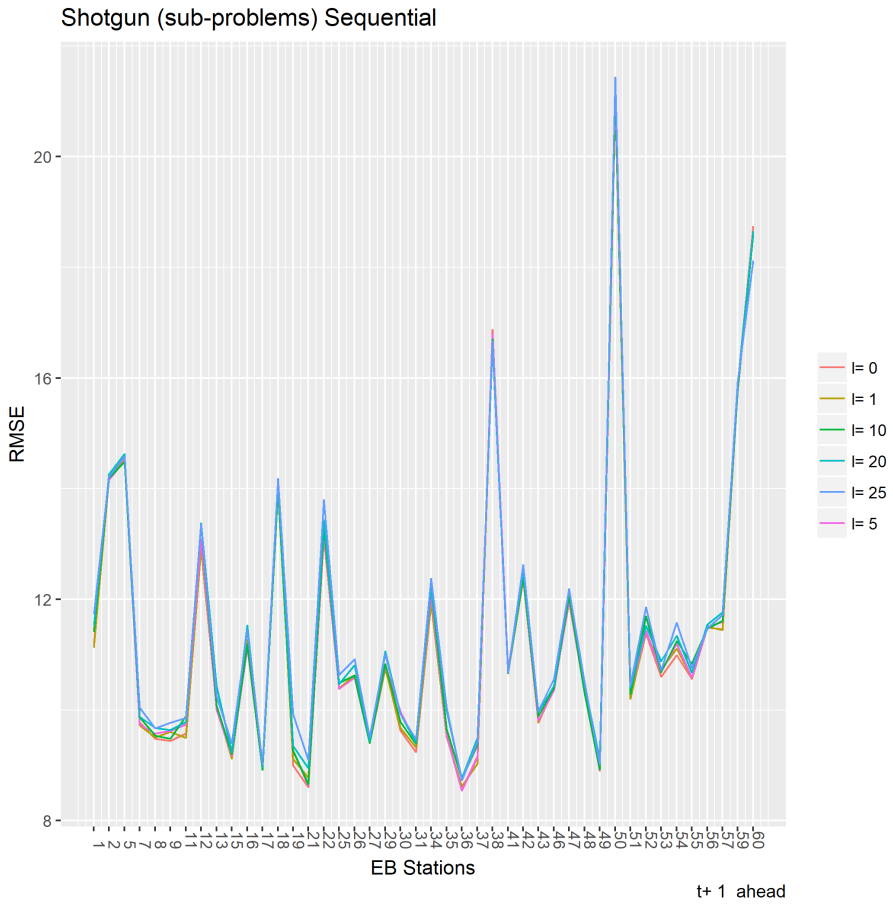


Figure B.1: Shotgun (sub-problem) sequential models RMSE for $t + 1$

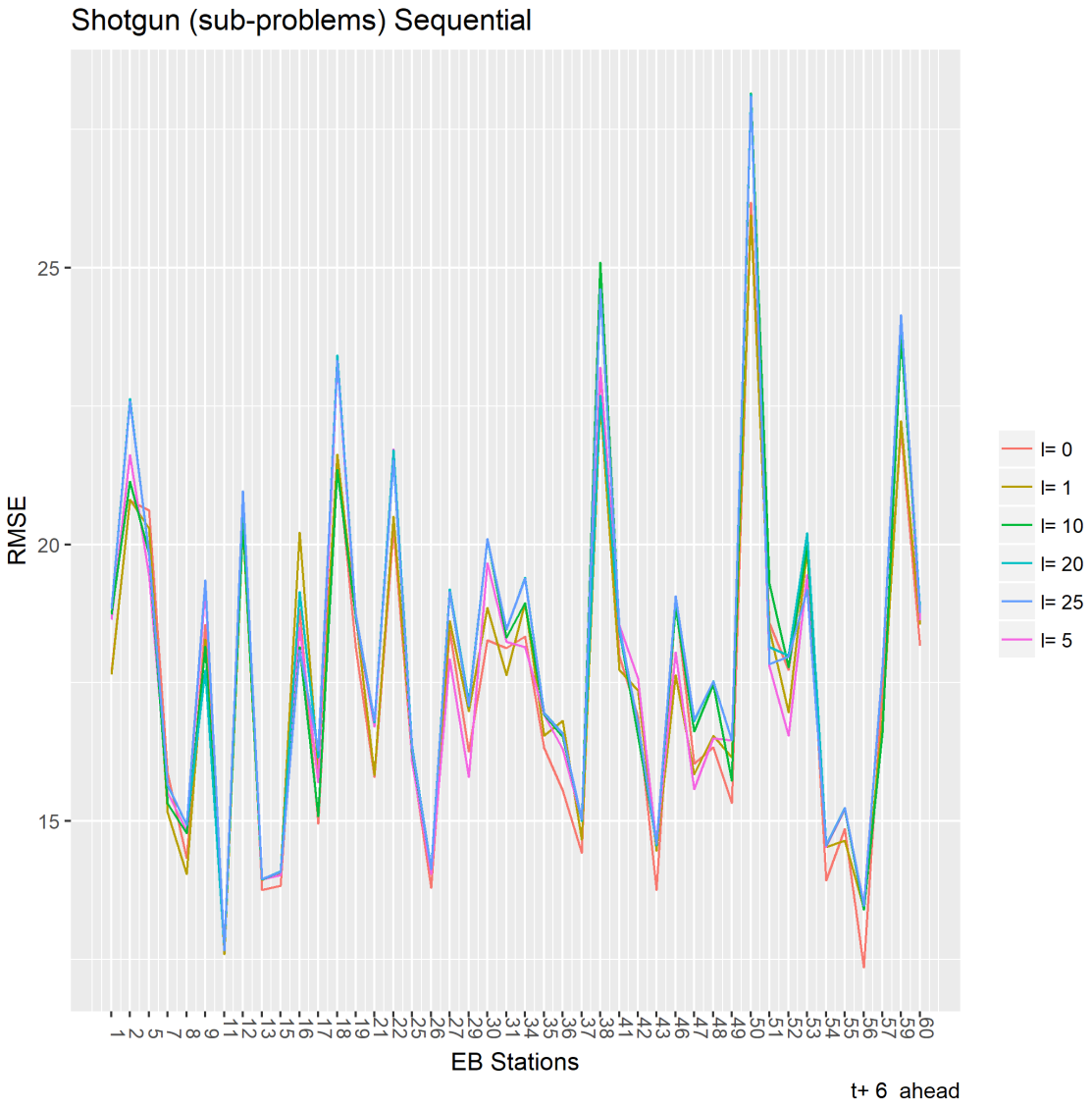


Figure B.2: Shotgun (sub-problem) Sequential models RMSE for $t + 6$

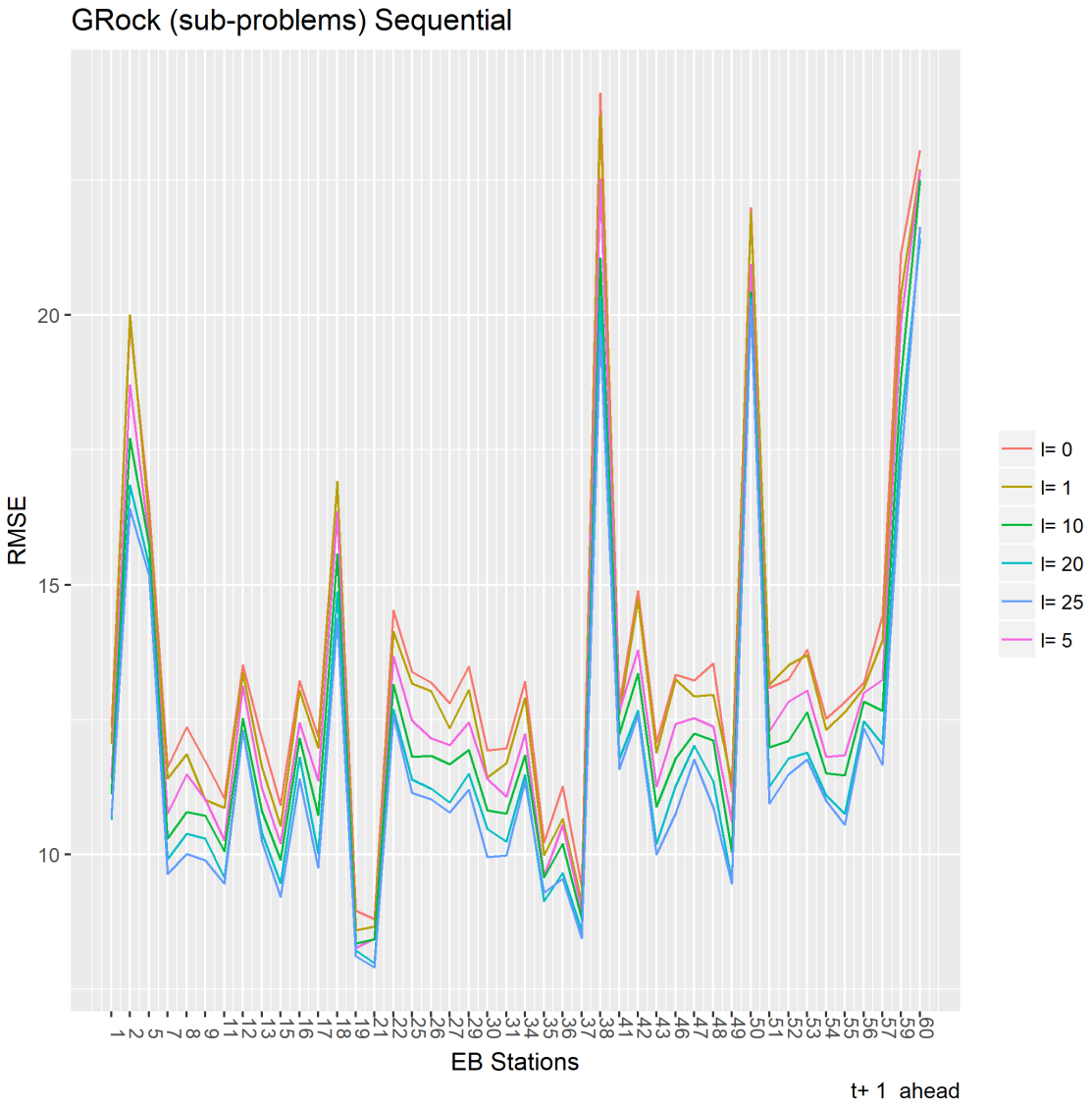


Figure B.3: GRock (sub-problem) parallel models RMSE for $t + 1$

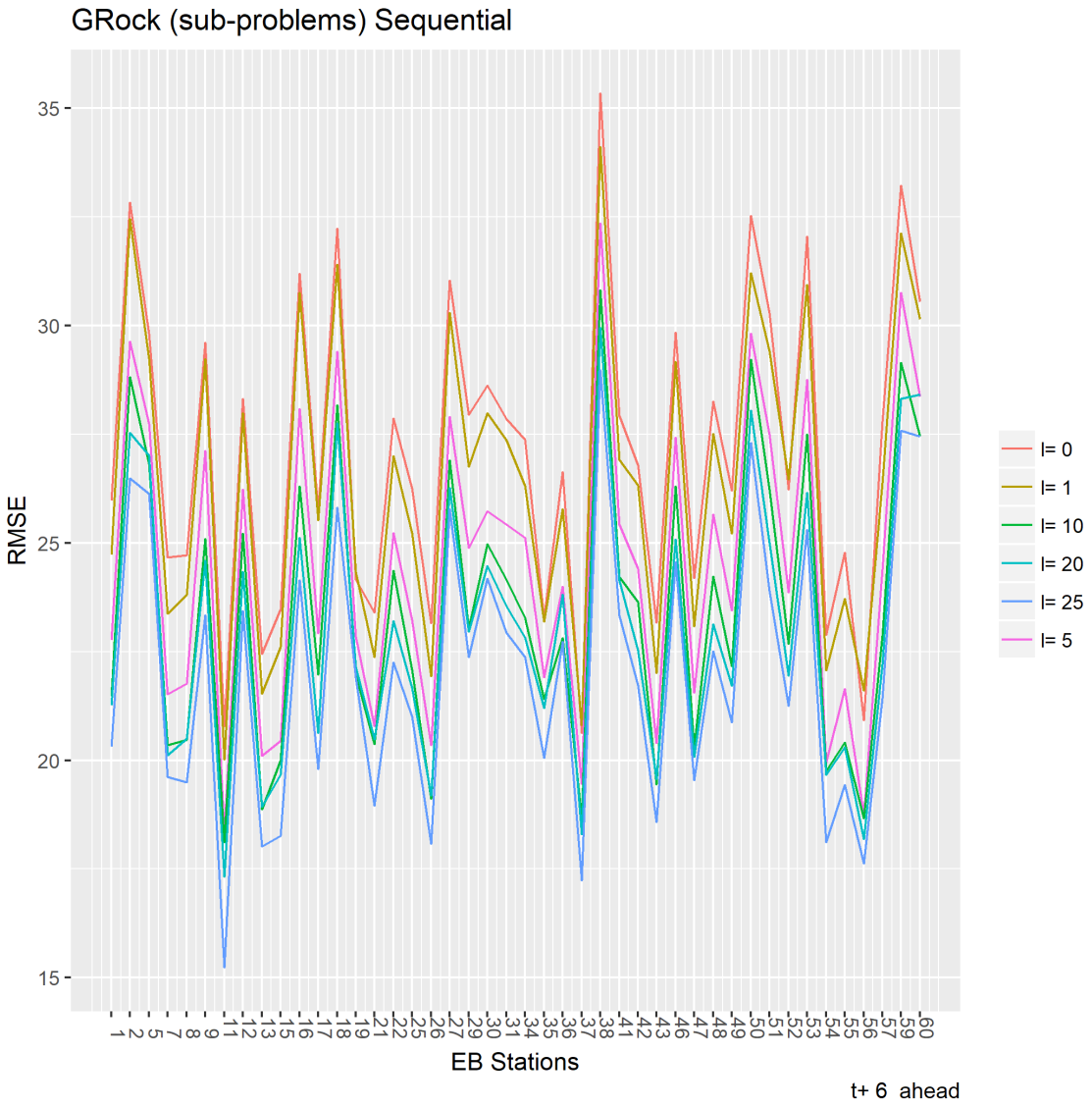


Figure B.4: GRock (sub-problem) parallel models RMSE for $t + 6$

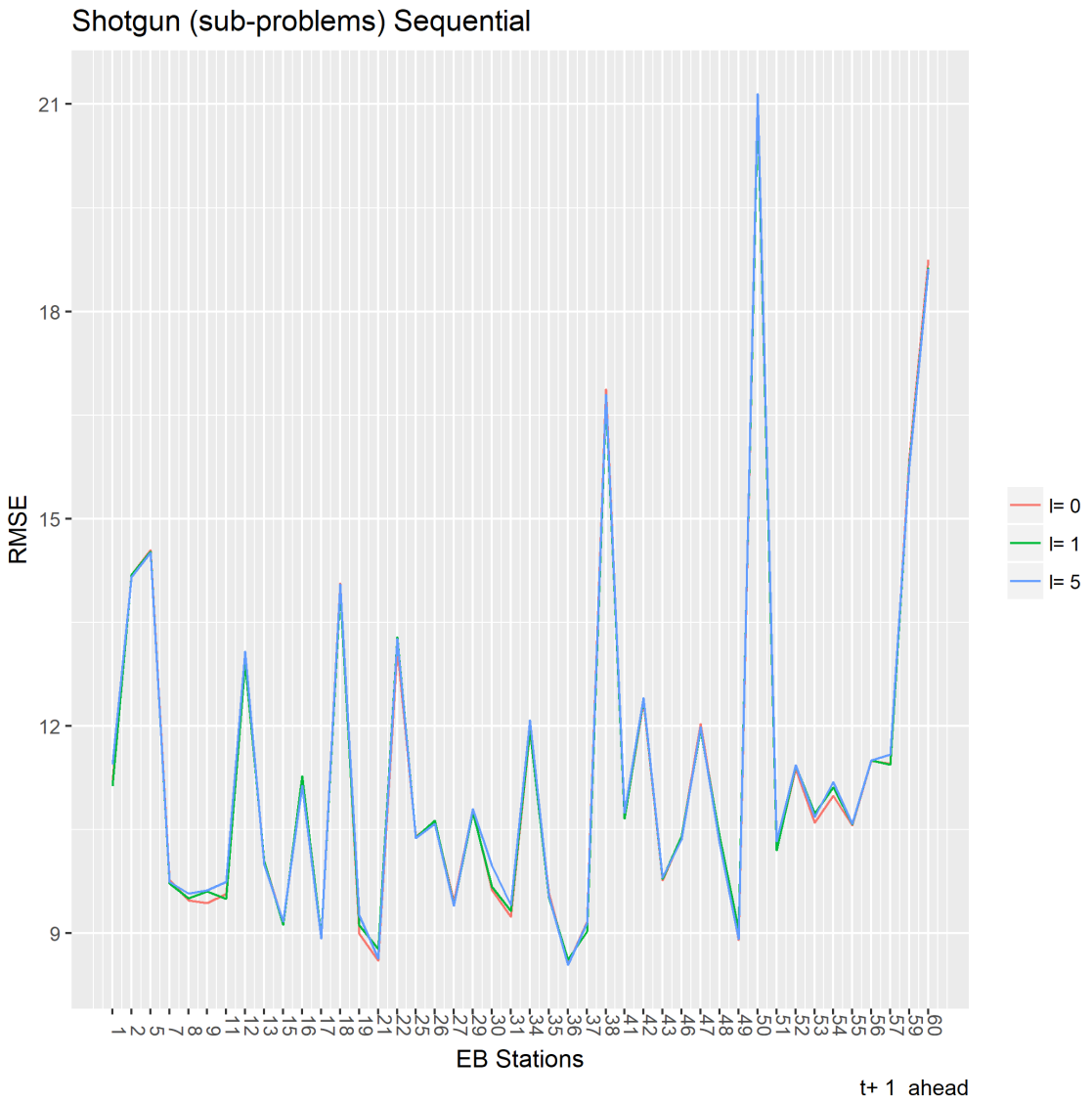


Figure B.5: Shotgun (sub-problem) sequential RMSE for step $t + 1$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$

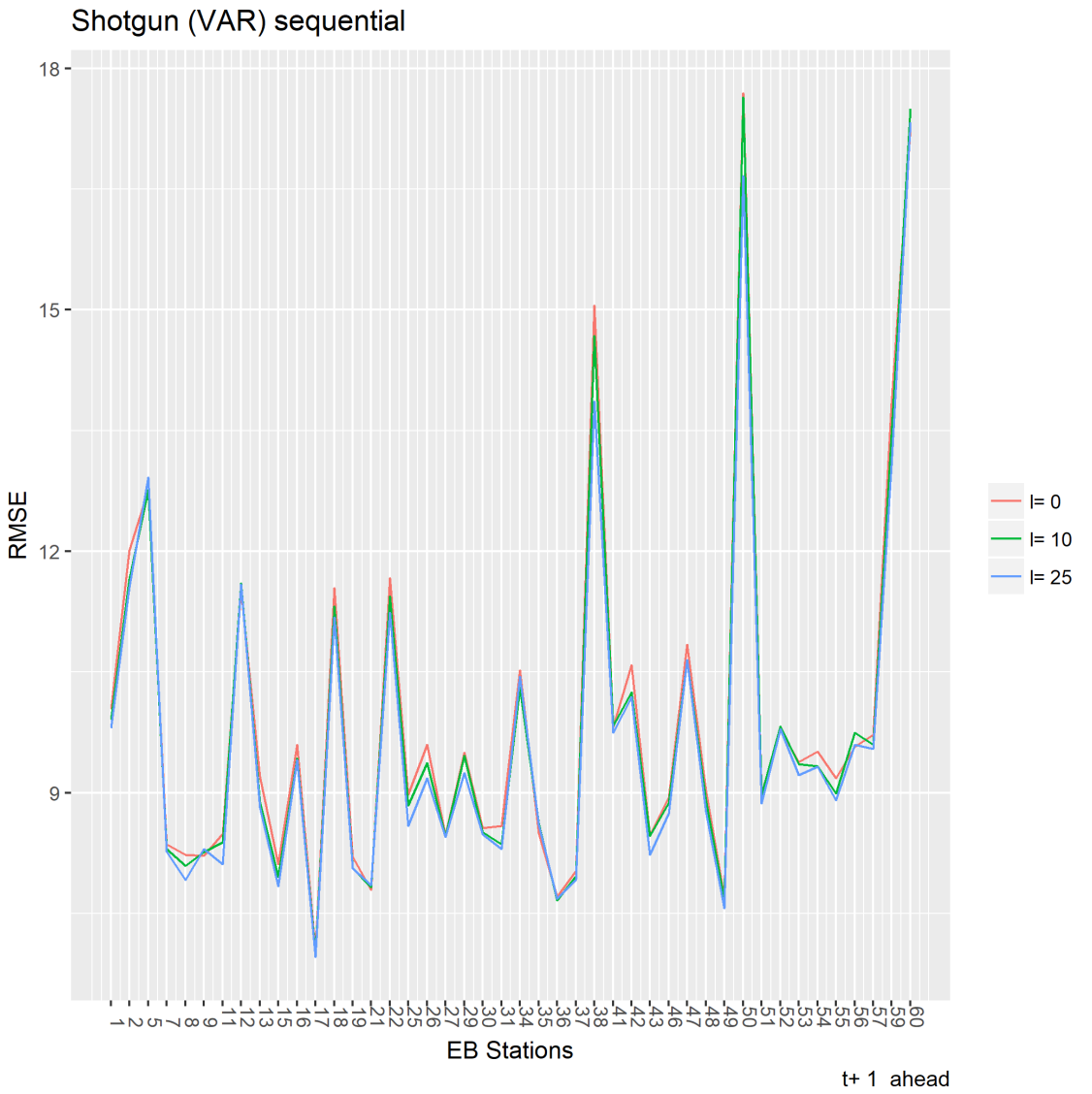


Figure B.6: Shotgun (VAR) sequential RMSE for step $t+1$. Models $\lambda = 0, \lambda = 10$ *and* $\lambda = 25$

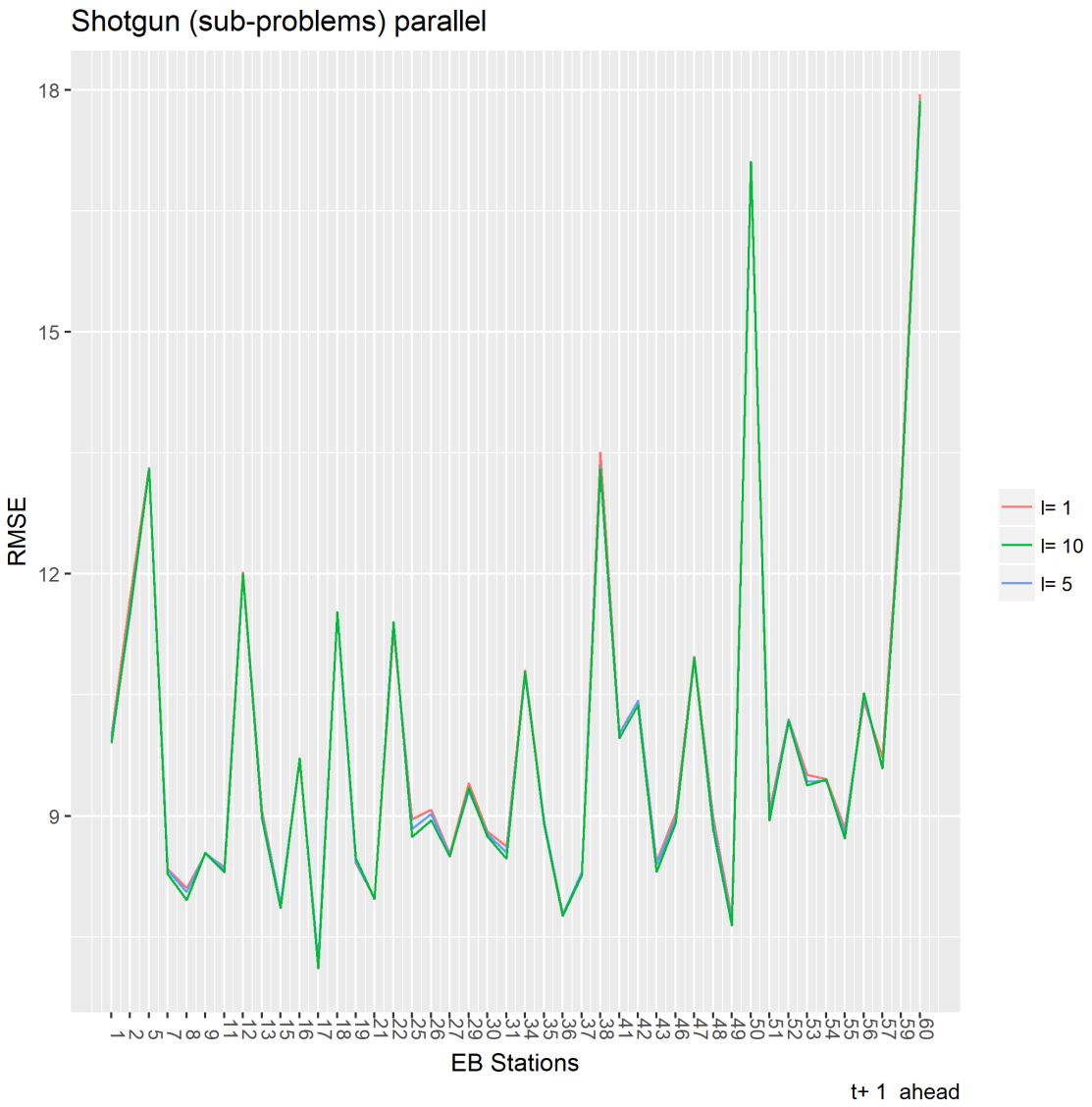


Figure B.7: Shotgun (Sub-problems) parallel RMSE for step $t + 1$. Models $\lambda = 0, \lambda = 10$ *and* $\lambda = 5$

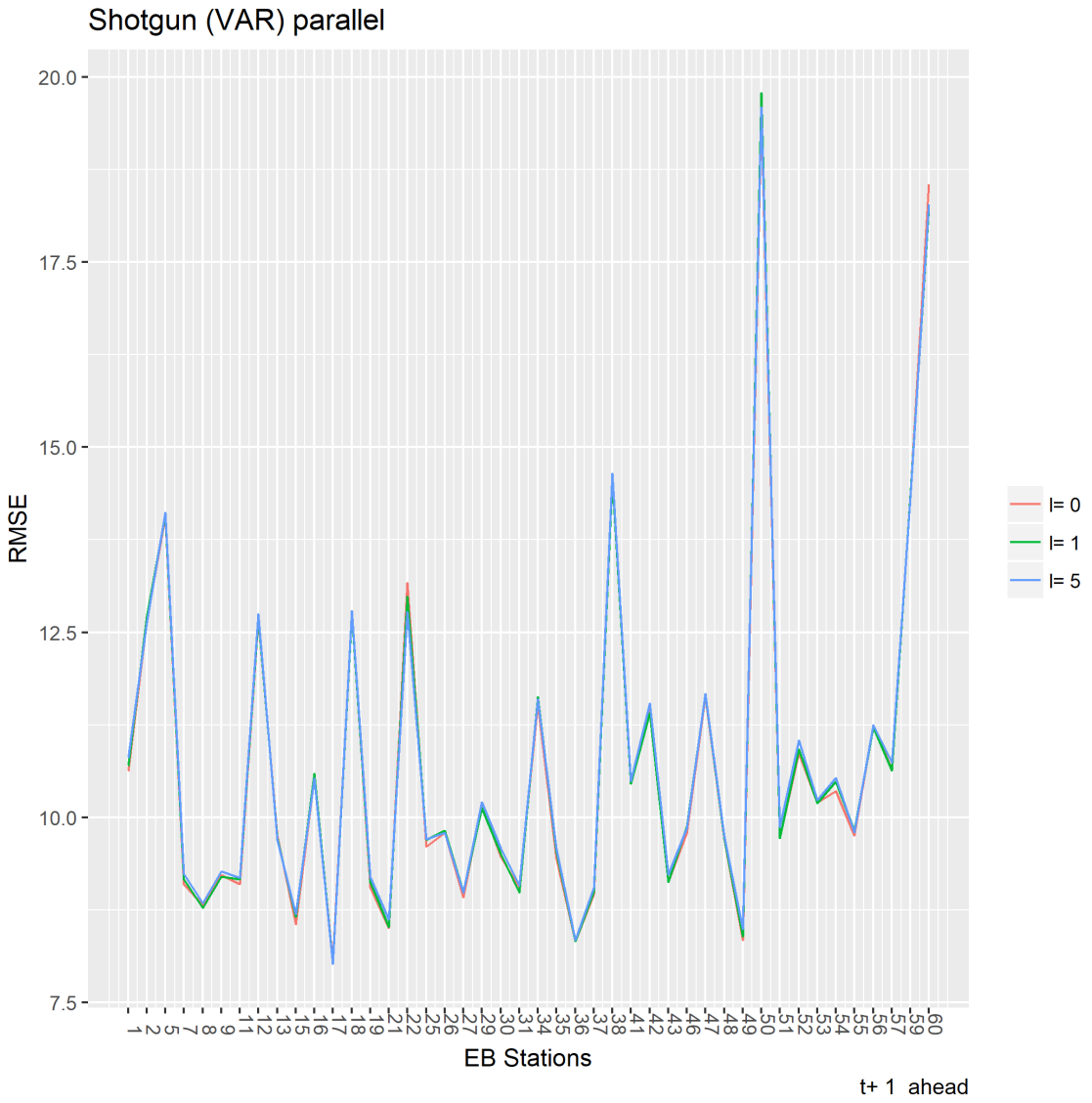


Figure B.8: Shotgun (VAR) parallel RMSE for step $t + 1$. Models $\lambda = 0, \lambda = 1$ *and* $\lambda = 5$

Appendix C

Forecast

In this appendix, we can see some extra examples of the forecast, for both approaches to the problem as well as the programming techniques applied.

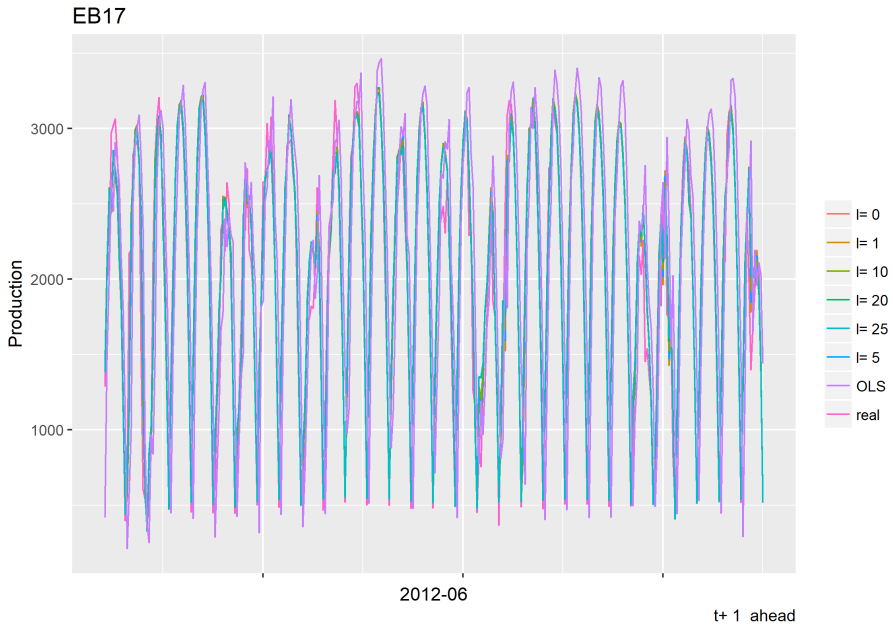


Figure C.1: EB17 Shotgun (sub-problem) parallel models'step $t + 1$ forecast for 2012-06

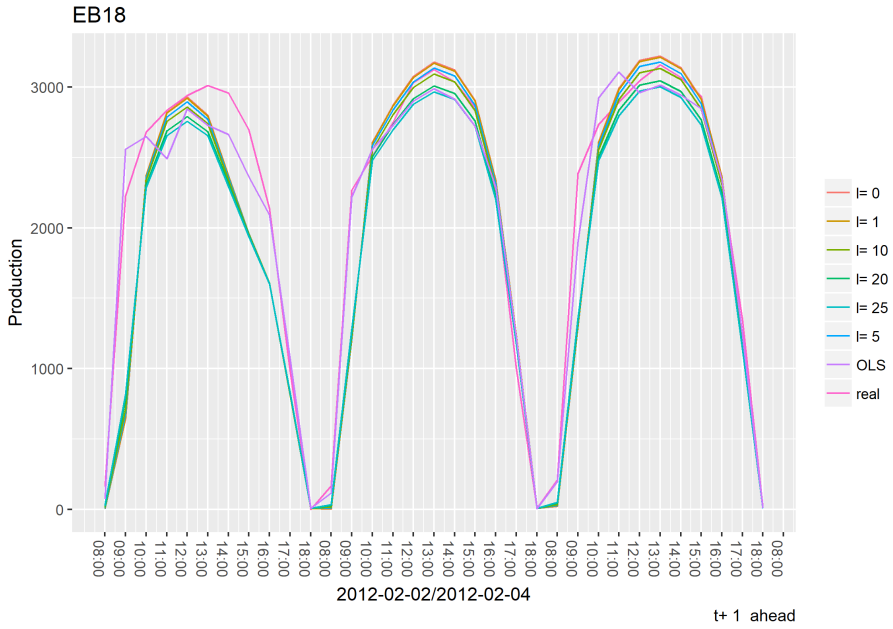


Figure C.2: EB18 GRock (sub-problem) Sequential models'step $t + 1$ forecast for period 2012-02-02/2012-02-04

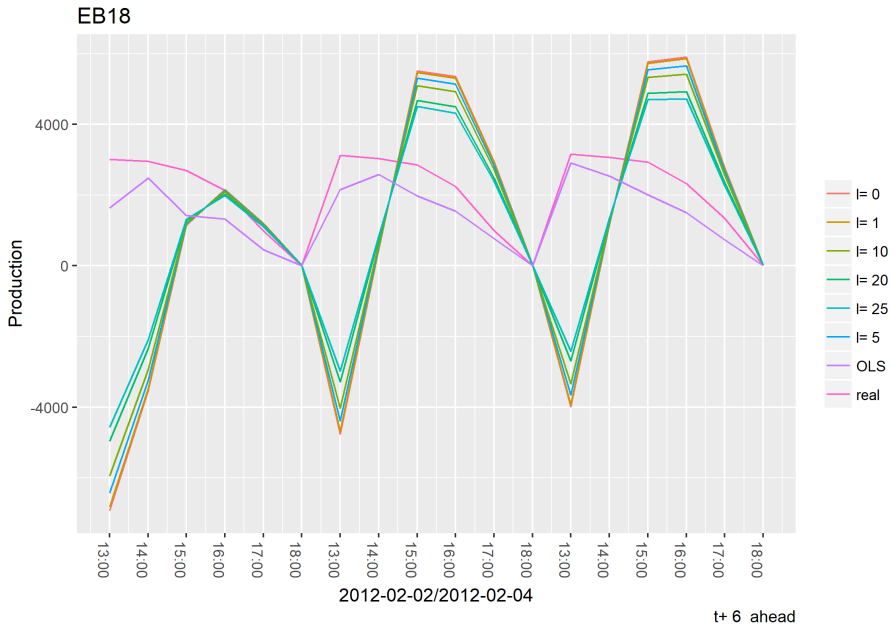


Figure C.3: EB18 GRock (sub-problem) Sequential models'step $t + 6$ forecast for period 2012-02-02/2012-02-04

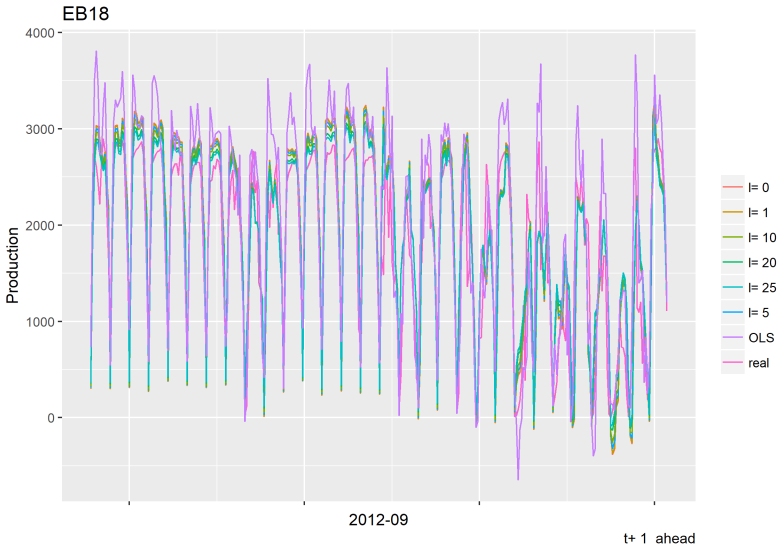


Figure C.4: EB18 GRock (VAR) Parallel models'step $t + 1$ forecast for 2012-09

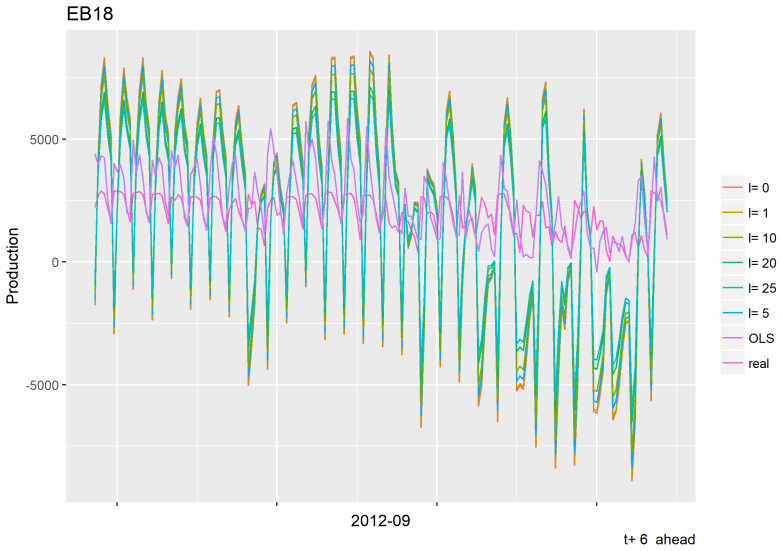


Figure C.5: EB18 GRock (VAR) Parallel models'step $t + 6$ forecast for 2012-09

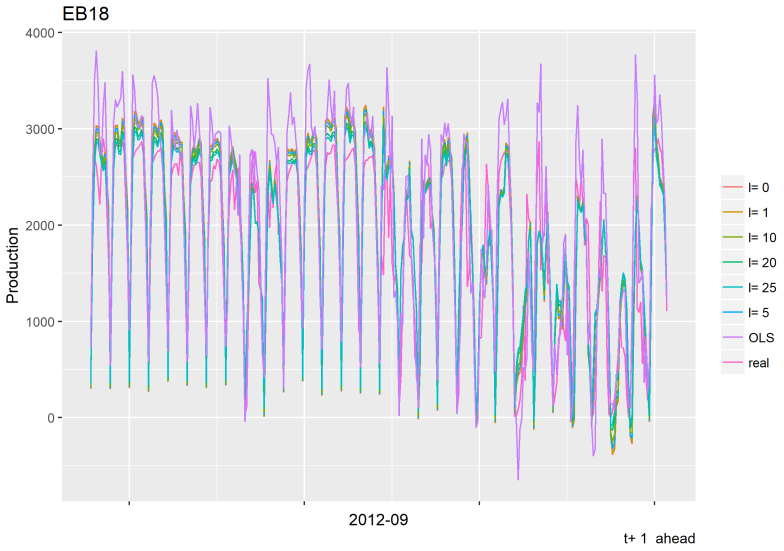


Figure C.6: EB18 GRock (VAR) Sequential models'step $t + 1$ forecast for 2012-09

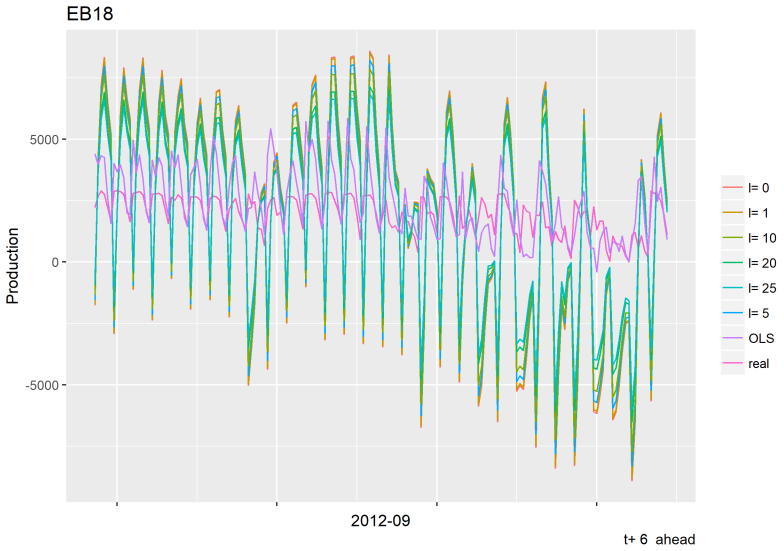


Figure C.7: EB18 GRock (VAR) Sequential models'step $t + 6$ forecast for 2012-09

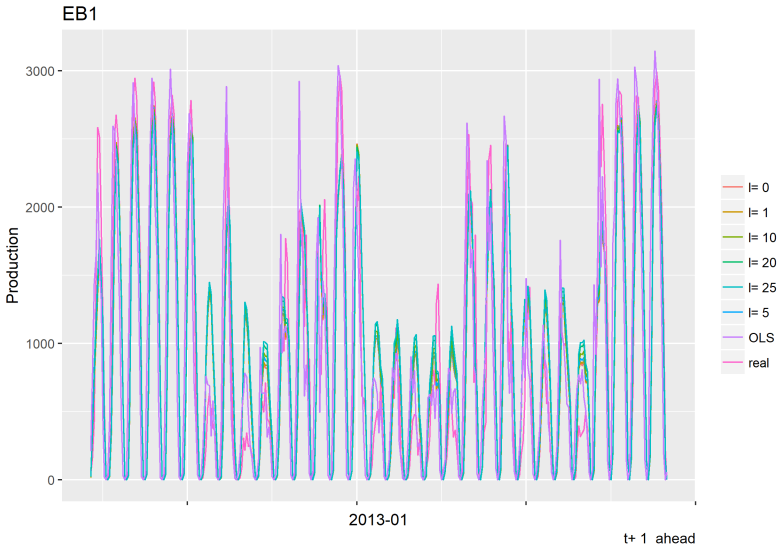


Figure C.8: EB1 Shotgun (VAR) Sequential models'step $t + 1$ forecast for 2013-01

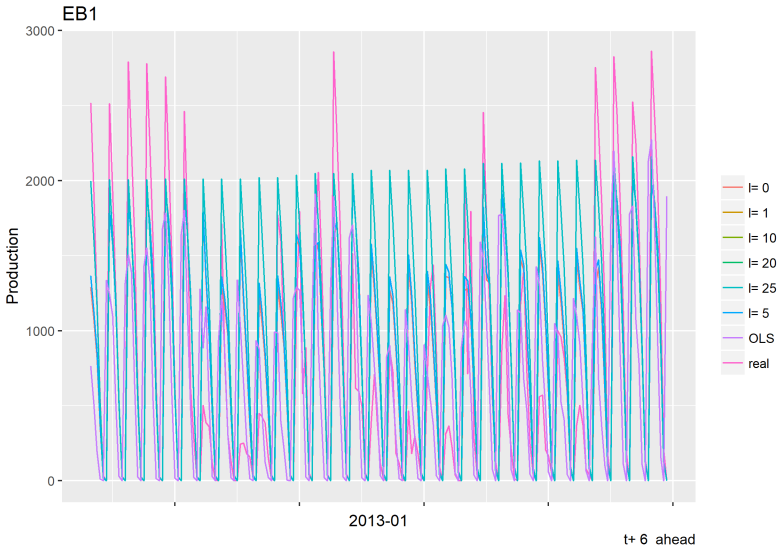


Figure C.9: EB1 Shotgun (VAR) Sequential models'step $t + 6$ forecast for 2013-01

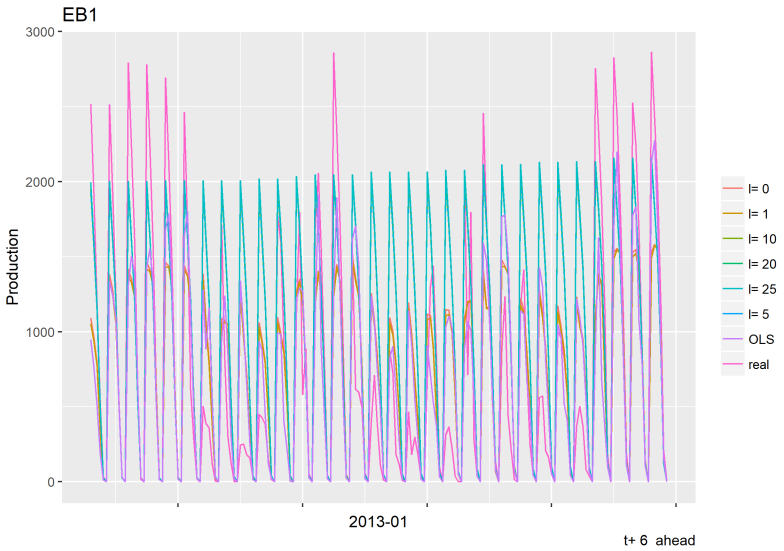


Figure C.10: EB1 Shotgun (VAR) Sequential models'step $t + 1$ forecast for 2013-01

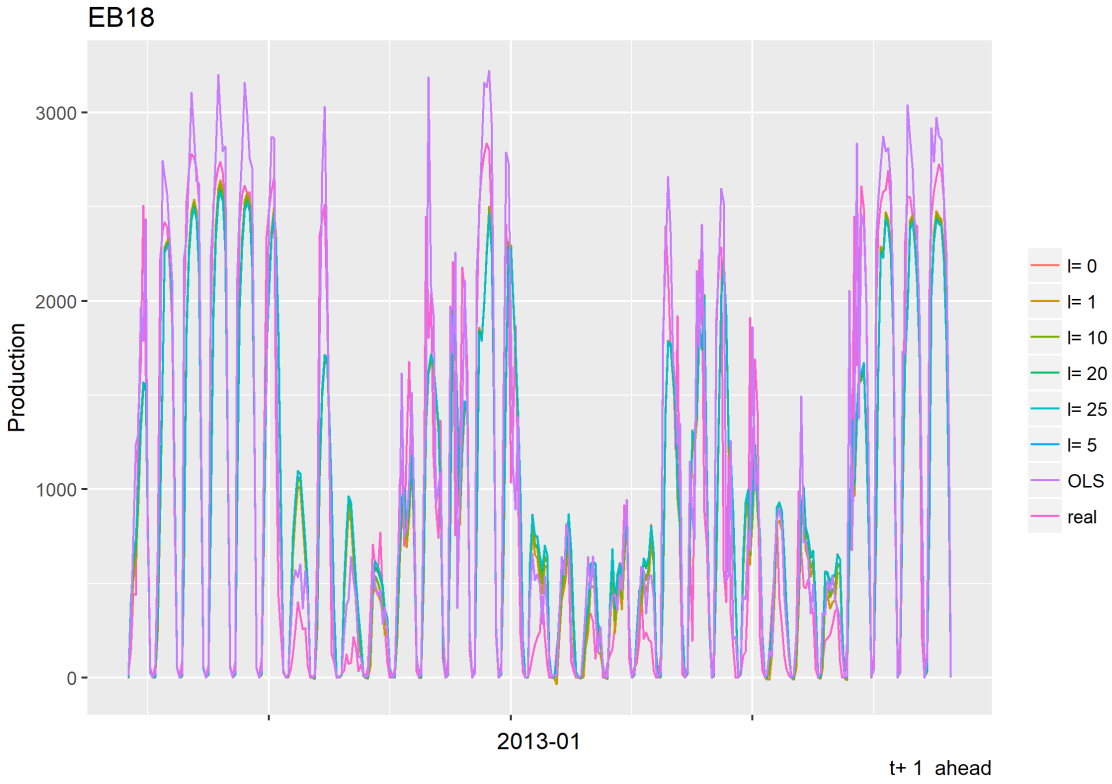


Figure C.11: EB1 Shotgun (VAR) Sequential models'step $t + 6$ forecast for 2013-01

Bibliography

Peder Bacher, Henrik Madsen, and Henrik Aalborg Nielsen. Online short-term solar power forecasting. *Solar Energy*, 83(10):1772 – 1783, 2009. ISSN 0038-092X. doi: http://dx.doi.org/10.1016/j.solener.2009.05.016. URL http://www.sciencedirect.com/science/article/pii/S0038092X09001364.

R.J. Bessa, A. Trindade, and V. Miranda. Spatial-temporal solar power forecasting for smart grids. *Industrial Informatics, IEEE Transactions on*, 11(1):232–241, Feb 2015. ISSN 1551-3203. doi: 10.1109/TII.2014.2365703.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. *CoRR*, abs/1105.5379, 2011. URL http://arxiv.org/abs/1105.5379.

J.C. Cao and S.H. Cao. Study of forecasting solar irradiance using neural networks with preprocessing sample data by wavelet analysis. *Energy*, 31(15): 3435 – 3445, 2006. ISSN 0360-5442. doi: http://dx.doi.org/10.1016/j.energy.2006.04.001. URL http://www.sciencedirect.com/science/article/pii/S0360544206001009. {ECOS} 2004 - 17th International Conference on Efficiency, Costs, Optimization, Simulation, and Environmental Impact of Energy on Process Systems17th International Conference on Efficiency, Costs, Optimization, Simulation, and Environmental Impact of Energy on Process Systems.

Jiacong Cao and Xingchun Lin. Application of the diagonal recurrent wavelet neural network to solar irradiation forecast assisted with fuzzy technique. *Engineering Applications of Artificial Intelligence*, 21(8):1255 – 1263, 2008. ISSN 0952-1976. doi: http://dx.doi.org/10.1016/j.engappai.2008.02.003. URL http://www.sciencedirect.com/science/article/pii/S0952197608000092.

Shuanghua Cao and Jiacong Cao. Forecast of solar irradiance using recurrent neural networks combined with wavelet analysis. *Applied Thermal Engineering*, 25: 161 – 172, 2005. ISSN 1359-4311. doi: http://dx.doi.org/10.1016/j.applthermaleng.

2004.06.017. URL http://www.sciencedirect.com/science/article/pii/S1359431104001814.

Volkan Cevher, Stephen Becker, and Mark Schmidt. Convex optimization for big data. *CoRR*, abs/1411.0972, 2014. URL http://arxiv.org/abs/1411.0972.

Chi Wai Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, and Byron Washom. Intra-hour forecasting with a total sky imager at the {UC} san diego solar energy testbed. *Solar Energy*, 85(11):2881 – 2893, 2011. ISSN 0038-092X. doi: http://dx.doi.org/10.1016/j.solener.2011.08.025. URL http://www.sciencedirect.com/science/article/pii/S0038092X11002982.

Hadja Maïmouna Diagne, Philippe Lauret, and Mathieu David. Solar irradiation forecasting: state-of-the-art and proposition for future developments for small-scale insular grids. In *WREF 2012-World Renewable Energy Forum*, 2012. URL https://ases.conference-services.net/resources/252/2859/pdf/SOLAR2012_0617_full%20paper.pdf.

Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27:65 – 76, 2013. ISSN 1364-0321. doi: http://dx.doi.org/10.1016/j.rser.2013.06.042. URL http://www.sciencedirect.com/science/article/pii/S1364032113004334.

Bihter Yerli Mustafa Kemal Kaymak Ahmet Duran Şahin Ercan İzgi, Ahmet Öztopal. Short?mid-term solar power prediction by using artificial neural networks. *Solar Energy*, 86(2):725 – 733, 2012. ISSN 0038-092X. doi: http://dx.doi.org/10.1016/j.solener.2011.11.013. URL http://www.sciencedirect.com/science/article/pii/S0038092X11004245.

Olivier Fercoq and Peter Richtárik. Accelerated, parallel and proximal coordinate descent. *CoRR*, abs/1312.5799, 2013. URL http://arxiv.org/abs/1312.5799.

Olivier Fercoq, Zheng Qu, Peter Richtárik, and Martin Takáč. Fast distributed coordinate descent for non-strongly convex losses. *IEEE Workshop on Machine Learning for Signal Processing, 2014*, 2014. URL http://arxiv.org/pdf/1405.5300v2.pdf.

Faranak Golestaneha, Pierre Pinson, and Hoay Beng Gooi. Generation and evaluation of space-time trajectories of photovoltaic power. October 2015.

Jing Huang and Matthew Perry. A semi-empirical approach using gradient boosting and k-nearest neighbors regression for gefcom2014 probabilistic solar power forecasting. *International Journal of Forecasting*, pages –, 2015. ISSN 0169-2070. doi: http://dx.doi.org/10.1016/j.ijforecast.2015.11.002. URL http://www.sciencedirect.com/science/article/pii/S0169207015001375.

Rich H. Inman, Hugo T.C. Pedro, and Carlos F.M. Coimbra. Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science*, 39(6):535 – 576, 2013. ISSN 0360-1285. doi: http://dx.doi.org/10.1016/j.pecs.2013.06.002. URL http://www.sciencedirect.com/science/article/pii/S0360128513000294.

Vijai Thottathil Jayadevan, Jeffrey J Rodriguez, Vincent PA Lonij, and Alexander D Cronin. Forecasting solar power intermittency using ground-based cloud imaging. 2012.

Wu Ji and Keong Chan Chee. Prediction of hourly solar radiation using a novel hybrid model of {ARMA} and {TDNN}. *Solar Energy*, 85(5):808 – 817, 2011. ISSN 0038-092X. doi: http://dx.doi.org/10.1016/j.solener.2011.01.013. URL http://www.sciencedirect.com/science/article/pii/S0038092X11000259.

Jakub Konečný, Zheng Qu, and Peter Richtárik. Semi-stochastic coordinate descent. *CoRR*, abs/1412.6293, 2014. URL http://arxiv.org/abs/1412.6293.

V.P.A. Lonij, V.T. Jayadevan, A.E. Brooks, J.J. Rodriguez, K. Koch, M. Leuthold, and A.D. Cronin. Forecasts of pv power output using power measurements of 80 residential pv installs. In *Photovoltaic Specialists Conference (PVSC), 2012 38th IEEE*, pages 003300–003305, June 2012. doi: 10.1109/PVSC.2012.6318280.

E. Lorenz and D. Heinemann. 1.13 - prediction of solar irradiance and photovoltaic power. In Ali Sayigh, editor, *Comprehensive Renewable Energy*, pages 239 – 292. Elsevier, Oxford, 2012. ISBN 978-0-08-087873-7. doi: http://dx.doi.org/10.1016/B978-0-08-087872-0.00114-1. URL http://www.sciencedirect.com/science/article/pii/B9780080878720001141.

E. Lorenz, J. Hurka, D. Heinemann, and H.G. Beyer. Irradiance forecasting for the power prediction of grid-connected photovoltaic systems. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 2(1):2–10, March 2009. ISSN 1939-1404. doi: 10.1109/JSTARS.2009.2020300.

Ricardo Marquez, Hugo T.C. Pedro, and Carlos F.M. Coimbra. Hybrid solar forecasting method uses satellite imaging and ground telemetry as inputs to {ANNs}. *Solar Energy*, 92:176 – 188, 2013. ISSN 0038-092X. doi: http://dx.doi.org/10.1016/j.solener.2013.02.023. URL http://www.sciencedirect.com/science/article/pii/S0038092X13000881.

Adel Mellit and Alessandro Massi Pavan. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected {PV} plant at trieste, italy. *Solar Energy*, 84(5):807 – 821, 2010. ISSN 0038-092X. doi: http://dx.doi.org/10.1016/j.solener.2010.02.006. URL http://www.sciencedirect.com/science/article/pii/S0038092X10000782.

C Monteiro, R Bessa, V Miranda, A Botterud, J Wang, G Conzelmann, et al. Wind power forecasting: state-of-the-art 2009. Technical report, Argonne National Laboratory (ANL), 2009.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. CORE Discussion Papers 2010002, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2010. URL http://EconPapers.repec.org/RePEc:cor:louvco:2010002.

W Nicholson, D Matteson, and Jacob Bien. Structured regularization for large vector autoregressions. 2014.

Zhimin Peng, Ming Yan, and Wotao Yin. Parallel and distributed sparse optimization. In *Signals, Systems and Computers, 2013 Asilomar Conference on*, pages 659–646. IEEE, 2013.

Zhimin Peng, Yangyang Xu, Ming Yan, and Wotao Yin. Arock: an algorithmic framework for async-parallel coordinate updates. 2015. URL http://arxiv.org/pdf/1506.02396.pdf.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: algorithms and complexity. *CoRR*, abs/1412.8060, 2014a. URL http://arxiv.org/abs/1412.8060.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation. *CoRR*, abs/1412.8063, 2014b. URL http://arxiv.org/abs/1412.8063.

P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *ArXiv e-prints*, December 2012.

Peter Richtárik and Martin Takác. On optimal probabilities in stochastic coordinate descent methods. *CoRR*, abs/1310.3438, 2013. URL http://arxiv.org/abs/1310.3438.

Mark Schmidt, Nicolas L. Roux, and Francis R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1458–1466. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4452-convergence-rates-of-inexact-proximal-gradient-methods-for-convex-optim
pdf.

Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l_1 regularized loss minimization. In *Proceedings of the 26th Annual International Conference on Machine*

Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009, pages 929–936, 2009. doi: 10.1145/1553374.1553493. URL http://doi.acm.org/10.1145/1553374.1553493.

Christopher Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980. URL http://EconPapers.repec.org/RePEc:ecm:emetrp:v:48:y:1980:i:1:p:1-48.

G.K. Singh. Solar power generation by pv (photovoltaic) technology: A review. *Energy*, 53:1 – 13, 2013. ISSN 0360-5442. doi: http://dx.doi.org/10.1016/j.energy.2013.02.057. URL http://www.sciencedirect.com/science/article/pii/S0360544213001758.

M.D. Tabone and D.S. Callaway. Modeling variability and uncertainty of photovoltaic generation: A hidden state spatial statistical approach. *Power Systems, IEEE Transactions on*, 30(6):2965–2973, Nov 2015. ISSN 0885-8950. doi: 10.1109/TPWRS.2014.2372751.

Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: Complexity and preconditioning. *CoRR*, abs/1304.5530, 2013. URL http://arxiv.org/abs/1304.5530.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Artur Jorge Teixeira Trindade. Very short-term solar power forecast for mv/lv distribution grids. Master's thesis, Faculdade de Engenharia da Universidade do Porto, March 2014.

A.G.R. Vaz, B. Elsinga, W.G.J.H.M. van Sark, and M.C. Brito. An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in utrecht, the netherlands. *Renewable Energy*, 85:631 – 641, 2016. ISSN 0960-1481. doi: http://dx.doi.org/10.1016/j.renene.2015.06.061. URL http://www.sciencedirect.com/science/article/pii/S0960148115300847.

Jianwu Zeng and Wei Qiao. Short-term solar power prediction using a support vector machine. *Renewable Energy*, 52:118 – 127, 2013. ISSN 0960-1481. doi: http://dx.doi.org/10.1016/j.renene.2012.10.009. URL http://www.sciencedirect.com/science/article/pii/S0960148112006465.