

# Dynamic Pooling for the Combination of Forecasts generated using Multi Level Learning

Silvia Riedel<sup>\*,\*\*</sup> and Bogdan Gabrys<sup>\*\*</sup>

**Abstract**—In this paper we provide experimental results and extensions to our previous theoretical findings concerning the combination of forecasts that have been diversified by three different methods: with parameters learned at different data aggregation levels, by thick modeling and by the use of different forecasting methods. An approach of error variance based pooling as proposed by Aiolfi and Timmermann has been compared with flat combinations as well as an alternative pooling approach in which we consider information about the used diversification. An advantage of our approach is that it leads to the generation of novel multi step multi level forecast generation structures that carry out the combination in different steps of pooling corresponding to the different types of diversification. We describe different evolutionary approaches in order to evolve the order of pooling of the diversification dimensions. Extensions of such evolutions allow the generation of more flexible multi level multi step combination structures containing better adaptive capabilities. We could prove a significant error reduction comparing results of our generated combination structures with results generated with the algorithm of Aiolfi and Timmermann as well as with flat combination for the application of Revenue Management seasonal forecasting.

## I. INTRODUCTION

### A. Motivation

In [1] and [2] we have provided a theoretical analysis of the behaviour of forecasts that have been diversified by three different methods: with parameters learned at different data aggregation levels, by thick modeling [3] and by the use of different function spaces. We have also mentioned that a side effect of the application of different types of diversification is that the number of forecasts to combine can get very big and that the resulting errors in the estimated covariance matrix can lead to high weight estimation errors [4]. We have therefore analysed the approach of error variance based pooling as proposed by Aiolfi and Timmermann [5] in order to handle that problem. We could show theoretically that we risk a significant loss in the expected forecast accuracy because of typical inhomogeneities in the covariance matrix for the analysed case. If covariance information is available in a sufficiently high quality, it is possible to run a clustering directly based on covariance information. We have considered a case when covariance information may not be available and proposed a new pooling approach that avoids the covariance inhomogeneities in considering only information

that is contained in a simplified covariance representation based on knowledge about the forecast generation process. One of the main advantages of the proposed approach is that the covariance matrix does not have to be calculated. We have compared the results of our approach with the approach of Aiolfi and Timmermann and explained the reasons for significant improvement of the proposed approach. Another advantage of our approach is that it leads to the generation of novel multi step multi level forecast generation structures that carry out the combination in different steps of pooling.

In this paper we present experimental results that support our theoretical findings. We also describe different evolutionary approaches in order to evolve the order of pooling of the dimensions. Extensions of such evolutions allow the generation of more flexible multi level multi step combination structures containing better adaptive capabilities. We compare the results of the algorithm of Aiolfi and Timmermann with flat combination and different combination structures that we have evolved for the application of Revenue Management seasonal forecasting.

In the next section we shortly summarize the two alternative pooling approaches. We start with a short summary of the approach of Aiolfi and Timmermann, describe then our alternative approach and finish with the extension of this approach in order to generate multi step combination structures. Section III provides then different versions of evolution of combination structures. After a short motivation we propose alternatives of how to determine the order of diversification dimensions used for pooling in our algorithm. Evolving that order avoids the time and cost consuming determination of the best structures based on static test data and allows additionally the adaptation to changed situations. Then we discuss different issues resulting from an analysis of the generated combination structures. Proposed solutions of these issues lead to the generation of much more flexible combination structures that can be obtained by some modifications in the proposed evolutionary algorithm. The last sections provide an introduction into the application, experimental results and conclusions.

## II. THE COMPARED POOLING APPROACHES

### A. The pooling approach of Aiolfi and Timmermann

In the context of forecast combination, Aiolfi and Timmermann [5] studied different approaches of individual forecast clustering connected with different combination models and trimming. They used quantiles and k-means clustering based on past forecast performance in order to find the optimal number of clusters and the optimal

\* Lufthansa Systems Berlin GmbH, Fritschestrasse 27-28, 10585 Berlin Germany, Phone: +49-30-34007141, Fax: +49-30-34007100, email: silvia.riedel@lhsystems.com.\*\* Computational Intelligence Research Group, School of Design, Engineering & Computing, Bournemouth University, Poole House, Talbot Campus, Poole, BH12 5BB United Kingdom, Phone: +44-1202-595298, Fax: +44-1202-595314 email: bgabrys@bournemouth.ac.uk

separation points between the forecast sets. All approaches analysed by Aiolfi and Timmermann run a clustering which is purely based on information about error variance terms. Correlation information is interpreted as inaccurate and not taken into account. We refer here to the algorithm which they called CEW in [5]. It generates a set of cluster forecasts  $\{\hat{y}^c\}$  based on a set of input forecasts  $\{\hat{y}\}$  with an algorithm that can be summarised as follows:

**Algorithm 1:**  $F^{cew}(\{\hat{y}\}) \rightarrow \{\hat{y}^c\}$

- 1) order  $\{\hat{y}\} \rightarrow \{\hat{y}_r\}$  depending on the ranks of forecast performance meaning the total error variances  $\delta^2$
- 2) determine  $I$  disjunct clusters  $c \subset \{\hat{y}\}$  by k-means clustering based on  $\delta^2$
- 3) remove the last cluster containing the worst forecasts (trimming)
- 4) for each cluster  $c$ : run a simple average combination  $F^{av}$  in order to achieve  $\hat{y}^c = F^{av}(c)$
- 5) apply an additional trimming of  $\{\hat{y}^c\}$ .

The results of the clusters  $\{\hat{y}^c\}$  are then combined using a linear combination model in order to achieve the total combined forecast  ${}^{comb}\hat{y} = F^{lin}(\{\hat{y}^c\})$ .

### B. Pooling based on the diversification

In [2] we have motivated theoretically why, seen from the covariance aspect, we should only cluster predictions which have been diversified by only one type of diversification. We have shown that choosing elements including more than one diversification contains the risk of inhomogeneities in the covariance matrix. This can lead to the application of suboptimal combination weights and a loss of relevant information.

Let a forecast generation space  $\mathcal{S} = \mathcal{D}_1 \times \dots \times \mathcal{D}_K$  be given with each dimension  $D$  representing one type of diversification. In order to avoid the problems caused by the mentioned inhomogeneities in the covariance matrix we have proposed to apply a pooling that is based on only one dimension  $D$  of  $\mathcal{S}$ .

The algorithm that we have proposed realizes a multi level fusion  $F^{ml}$  of a set of forecasts  $\{\hat{y}\}$  into a set of clustered forecasts  $\{\hat{y}^c\}$  corresponding to one diversification dimension  $D$ . We pool forecasts that differ corresponding to  $D$ , but are not diversified corresponding to all of the other diversification dimensions. In order to avoid too much variations in the error variances in such sets of forecasts we follow a trimming strategy and eliminate all those forecasts with relatively bad quality related to that dimension. We discard, e.g., the obviously bad parameter values for a given model at a given level or predictions at completely instable levels for a given model with given parameter settings.

**Algorithm 2:**  $F^{ml}(\{\hat{y}\}, D) \rightarrow \{\hat{y}^c\}$

- 1) define cluster  $c$  corresponding to each element  $s \in \mathcal{S}/\{D\}$
- 2) remove the worst forecasts in each cluster  $c$  by trimming

- 3) for each cluster  $c$  run a linear combination  $F^{lin}$  in order to achieve the forecast  $\hat{y}^c = F^{lin}(c)$

The forecast generation space corresponding to the generated clusters  $\{\hat{y}^c\}$  is  $\mathcal{S}/\{D\}$  meaning  $\mathcal{S}$  reduced by dimension  $D$ . Depending on the strength of the used trimming strategy the *simple average model* or the *optimal model with assumption of independence* [4] can be chosen as combination model  $F^{lin}$ .

The trimming that we use is guided by the total error variance of the best forecast per cluster. All forecasts of which the total error variance differs more than  $\lambda\%$  of that of the best forecast of the cluster are removed. The number of accepted predictions per pool is restricted as well. Details related to the definition of the forecast generation space and the algorithm as well as an example can be found in [2].

### C. Generation of multi step combination structures

As we have just mentioned, the result of a pooling related to a dimension  $D$  is again a set of forecasts the generation of which can be described by the forecast generation space  $\mathcal{S}/\{D\}$ . If  $\mathcal{S}/\{D\}$  contains only one element (meaning all existing dimensions have already been aggregated or have the range 1), the pooling has generated a final result which can be used as the final combined forecasts. Otherwise we can combine the remaining forecasts using a flat combination as do Aiolfi and Timmermann.

But as the number of resulting forecasts can still be big, there is the other option to repeat the pooling approach based on  $\mathcal{S}/\{D\}$  with a chosen dimension  $D \neq D$ . This idea leads to an approach of the successive generation of pools and so the generation of multi step combination structures. Each step leads to the reduction of one dimension of  $\mathcal{S}$  so that the total number of steps is defined by the dimensionality of  $\mathcal{S}$ .

Figure 1 shows an extract of the resulting structure for an example containing diversification concerning the use of two levels, four methods and two parameters controlled by thick modeling. We get  $\mathcal{S} = D_1 \times D_2 \times D_3 \times D_4 = [0, 1] \times [0, \dots, 3] \times [0, \dots, 10] \times [0, \dots, 8]$ . In this example we have used the trimming approach of selecting not more than the best three forecasts per pool.

## III. EVOLVING MULTI STEP MULTI LEVEL COMBINATION STRUCTURES

### A. Motivation for using evolutionary computation

Even if good experimental results show that the use of multi step structures may be a way to overcome the problems of combining a big number of forecasts, the approach of using such predefined multi step structures is limited in the sense that it needs a lot of expert knowledge in order to identify promising ones. Potential structures, once identified, have to be verified by experiments using try and error principles. And as the fixed structures contain only limited adaptive capabilities, they would have to be rebuilt on a regular basis.

The best structures do not necessarily need to be the intuitive ones. Often solutions found in nature do not follow our

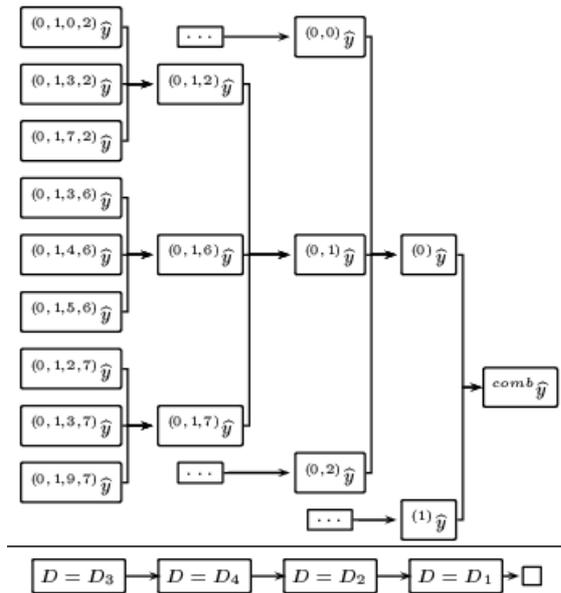


Fig. 1. Extract of a more complex combination structure with  $\mathcal{S} = D_1 \times D_2 \times D_3 \times D_4 = [0, 1] \times [0, \dots, 3] \times [0, \dots, 10] \times [0, \dots, 8]$ . Below the line it is indicated which dimension  $D$  has been chosen in each step.

intuition. All these reasons motivate the search for dynamic approaches generating and adapting structures automatically. The generated optimal structures need to be able to work well in a changing environment. Evolutionary computation offers common algorithms to solve such kind of problems. In [6] we have already presented evolutions of multi level multi step combination structures. We have used genetic algorithms and evolutionary programming in order to evolve combination structures purely based on the total error variances. Now we present alternative evolutions that consider our theoretical findings presented in [2].

### B. Evolving the order of dimensions used for pooling

We have seen that algorithm  $F^{ml}$  needs in each step the information which dimension  $D$  is used for the next step of pooling. In the example shown in figure 1 first we combine dimension  $D_3$ , than dimension  $D_4$  and so on.

The question of which dimension to choose next is a crucial task. As we assume covariance information as not reliable we do not have the needed information in order to make this decision on a theoretical basis. Even if we had reliable covariance information, we could determine the ‘ideal’ dimension for pooling only for the next step of pooling. We would need to have higher order statistics information in order to know the correlation of forecasts representing the results of the first step of pooling based on the original forecasts. This means that even with reliable covariance values there is the need for a very time expensive recalculation of covariances after each step of pooling. An alternative option is the determination of the order of dimensions used for pooling using an evolutionary strategy. We will now describe different options of how we can define

such a kind of evolution.

1) *Description of genes and chromosomes:* Let’s assume we have a forecast generation space given by  $\mathcal{S} = \mathcal{D}_1 \times \dots \times \mathcal{D}_K$  with  $K$  the number of diversified dimensions as already described above. The generation of combination structures following algorithm  $F^{ml}$  is determined by a vector that indicates the order of the dimensions  $D$  to be used for pooling.

We define genes as  $g \in \mathcal{N}$ . They represent each an index  $k$  of a dimension of the forecast generation space. Chromosomes are defined as vectors of disjunct genes  $cr \in \mathcal{N}^K$ . The order of the genes in a chromosome describes the order dimensions used for pooling. The example for a chromosome  $cr \in \mathcal{N}^4$  corresponding to the pooling described in figure 1 is provided in figure 2 as parent1.

2) *The fitness:* The most simple and intuitive criterion to optimize is defined by the quality of the resulting forecasts. We want to learn combination structures which generate high quality combined predictions measured on unseen data. The fitness is calculated as a mean absolute deviation value on the level of forecasting and is given as

$$f = \frac{1}{t_2 - t_1 + 1} \sum_{t=t_1}^{t_2} |^{comb} \hat{y}_t - y_t|, \quad (1)$$

where  $t$  represents a time index over the evaluation period  $[t_1, t_2]$ .

3) *Crossover:* Two types of child generation have been used in separate experiments.

The first type generates a child based on two parent elements. The crossover considers the position of the dimensions in the chromosomes of the two parents. The child is calculated using the following algorithm :

- initialise the child  $cr^{child}$  without any genes
- loop  $k = 1$  to  $K$ 
  - select randomly one of the parents  $cr^{p1}$
  - if gene  $cr_k^{p1}$  is not yet contained in the child  $\rightarrow$  add gene  $cr_k^{p1}$  to the child
  - if gene  $cr_k^{p2}$  of the other parent is not yet contained in the child  $\rightarrow$  add gene  $cr_k^{p2}$  to the child

An Example of two parents with a generated child is shown in figure 2.

The second crossover uses only one parent element. The child is generated by an exchange of any randomly selected gene with a neighbored gene. If we accept the child only if it performs better than its parent similar to Tabu Search, this type of evolution can be carried out with a very small population or even a single chromosome. Figure 2 provides also an example for this type of crossover.

4) *Mutation:* The mutation has been used in order to adapt the trimming percentage. We have carried out a mutation in each fifth crossover. During the mutation the trimming percentage  $\lambda$  has been randomly modified up to 10 percent of the previous value. We have experimented with two types of representation of  $\lambda$ : a global representation with the same value used for all steps of pooling and a separate representation per combined pool.

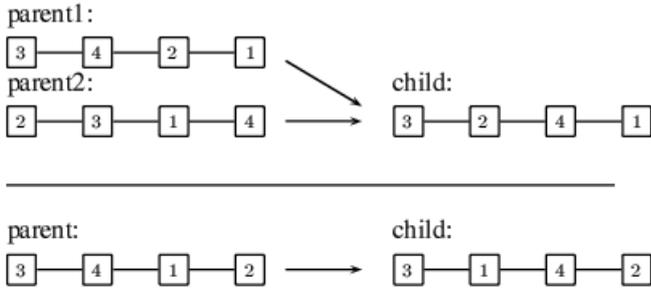


Fig. 2. Example for two types of crossover. The first crossover shown above the line is based on two parents, the second one on a single parent element.

5) *Observed behaviour and resulting issues:* While the choice of the crossover operator did not have a big impact on the out of sample quality of the resulting structures, the evolved combination structures performed very different corresponding to the selected mutation. An analysis of the resulting structures showed that both approaches of mutation cause a different type of problems.

The use of a separate representation per combined pool of  $\lambda$  generated structures that suffer from overfitting. The big number of evolved parameters (one per pool) led to very flexible structures that have been strongly influenced by the noise in the historical data.

The use of a single parameter for the whole structure did not allow a sufficient adaptation in relation to the different types of diversification. While in the case of thick modeling extreme parameter values lead to the generation of bad quality forecasts that can be trimmed without any problem (and need to be trimmed), the situation can be different for the use of different levels or methods. Forecasts that perform worse than the best forecast per pool can contain divers information and therefore be relevant. The result of this type of mutation have been stable and well performing combination structures, but often relevant information got lost because of trimming.

*Therefore there is a need for flexible and stable combination structures that respect the homogeneity of the covariance matrix, but trimming of relevant information is avoided.*

### C. Evolving more dynamic pooling structures

1) *Extension 1: Additional pooling of elements of diversification dimensions:* The first approach in order to fulfil these needs is based on the idea of additional poolings of elements of a dimension. The subsets are then treated like separate diversification dimensions.

Example : We split dimension  $D_1$  into two pools  $D_{1,1}$  and  $D_{1,2}$ . The split is based on the total error variances following e.g. the approach of k-means clustering as proposed by Aiolfi and Timmermann. We achieve a bigger total number of pools of forecasts that are less divers concerning the total error variance. The splitting is carried out as a mutation during the evolution. Each dimension is split only once.

The approach has the following positive effects: (a) there is a bigger set of resulting combination structures and with that a higher flexibility, (b) as also in this approach we combine in each step elements differing only concerning one diversification we respect the homogeneity of the covariances and (c) the splitting has an impact on the trimming because smaller pools of comparable forecasts cause less trimming

Example for mutation:



The pooling of the split dimensions does not necessary need to be carried out as a sequence. During the following evolution the order of the pooling of the parts can be varied in relation to other dimensions.

This enables the generation of a large variety of combination structures. An example for a resulting structure is shown in figure 3.

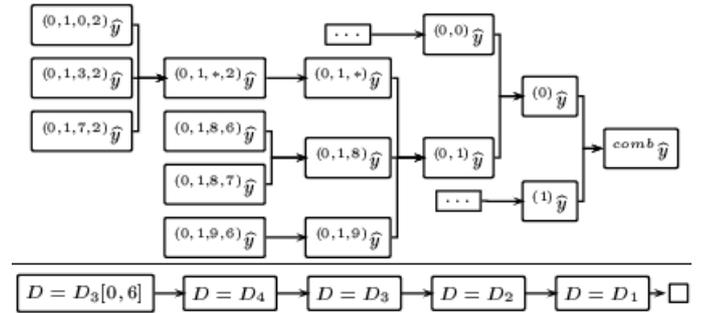


Fig. 3. Extract of a more complex combination structure with  $S = D_1 \times D_2 \times D_3 \times D_4 = [0, 1] \times [0, \dots, 3] \times [0, \dots, 10] \times [0, \dots, 8]$ . Dimension 3 has been split into subsets of elements  $[0, 6]$  and  $[7, 10]$ . '\*' stands for an aggregated subset of a dimension. Below the line it is indicated which dimension  $D$  has been chosen in each step.

2) *Extension 2: Element-specific order of pooling:* The pooling approaches as described above consider the same order of pooling for all elements. If we have e.g. combined the elements corresponding to pooling in relation to a first dimension and we achieve 20 pools representing each combination of elements of the resulting dimensions, we would choose the same dimension for the next step of pooling for each resulting pool. This dimension is not necessarily the best choice for each of the elements. We can generate more flexible combination structures by making the choice in relation to the resulting pool.

The chromosomes are now represented as trees of genes in which each node corresponds to the index of the next diversification dimension to be combined, the branches correspond to the resulting pools. Figure 4 shows an example. For the child generation we have used an approach that represents an extension of the child generation based on a single parent. We randomly select a gene and exchange it with all of its children.

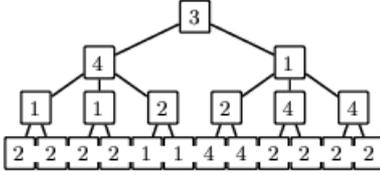


Fig. 4. An example of a chromosome with element-specific order of pooling

#### IV. EXPERIMENTS

In [6] we have described a testbed based on which we have carried out first experiments concerning the evolution of multi level multi step combination structures for the application of seasonal forecasting in Revenue Management for Airlines. The chosen testbed for our new experiments included booking data of the same origin-destination pairs (ODs) of a major German carrier. We compared the described new approaches with the approach of Aiolfi and Timmermann as well as the existing commercial forecasting system. Data from 2001 to October 2006 has been splitted into the following sets: the years 2001 to 2003 have been used for learning the parameters of to the individual forecasts, years 2004 and 2005 have been used as training set for the evolution, 2006 represents the test set. We have used a population size of 20 chromosomes, a probability of crossover of 0.7 and a probability of mutation of 0.05.

##### A. Description of the Application: Methods

In O&D Revenue Management Systems [7] seasonal predictions have to be carried out at a very fine level where the behaviour changes very quickly so that it is not possible to take a large number of historical data into account[8]. Predictions have to be generated not only for different flights or origin-destination-itinerary pairs (the so called ODIs), but also separately for different fareclasses (F) representing different prices and booking restrictions as well as different point of sales (POS). Lets assume we have to model a seasonal dependency of the booking behaviour on the calendar week in terms of seasonal factors <sup>1</sup>. In our example  $t$  represents a weekly time index, the variable  $x_t \in [1, 53]$ ,  $b_t$  the number of bookings achieved in week  $t$ ,  $y_t$  a seasonal factor representing the deviation between bookings in the given calendar week and the total booking expectation  $y_t = \frac{b_t}{E(b)}$ .

Figure 5 shows  $y$  depending on  $x$  together with two examples for learned seasonal factors. They are both based on estimations of the seasonal factors  $s_x^{hist}$  per year

$$s_x^{hist} = \min(\max(\frac{1}{2\Phi_I + 1} \sum_{i=-\Phi_I}^{\Phi_I} [y_{x+i}], \Phi_{low}), \Phi_{high}). \quad (2)$$

<sup>1</sup>Of course there are other (seasonal) impacts like e.g. day of week dependencies. We will restrict our argumentation to dependencies on the calendar week and assume no other known impacts to the booking behaviour in order to keep the example simple.

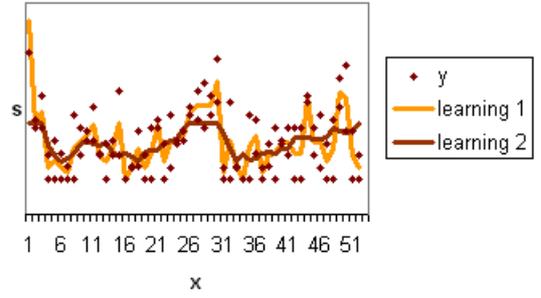


Fig. 5. Measured seasonal factors during 2 years with two learned curves  $s_x^{hist}$ . Learning 1 is carried out with a set of parameters that allow a very high flexibility:  $\Phi_{low} = 0$ ,  $\Phi_{high} = 5$  and  $\Phi_I = 0$ . Learning 2 is carried out with parameters that generate a more stable curve:  $\Phi_{low} = 0.5$ ,  $\Phi_{high} = 2$  and  $\Phi_I = 2$ .

which have then been averaged over the two years in order to represent an estimation for the total historical behaviour.

The two examples of learning the seasonal behaviour differ concerning the used parameters  $\Phi_I$ ,  $\Phi_{low}$  and  $\Phi_{high}$ . Parameter  $\Phi_I$  represents the size of the neighbourhood of a calendar week that is taken into account for the estimation of the seasonal behaviour. A bigger value means a noise reduction and the generation of smoother seasonal curves. But it also represents a restriction in modelling quick changes in the seasonal behaviour between neighbored weeks. The other two parameters  $\Phi_{low}$  and  $\Phi_{high}$  are also used for stabilisation purposes. They represent a lower and an upper limit on the expected seasonal factors. Strong restrictions again mean a noise reduction and allow for instance to avoid a zero season assumption in case of no historical bookings measured at the ODIFPOS level for a given calendar week, but represent also a restriction in flexibility of the learned seasonal factors.

The learned seasonal factors can be used in order to generate predictions for future seasonal behaviour. We have to consider already measured bookings  $b_\tau$  for a future departure  $t$  that should be predicted,  $\tau$  representing a given number of days prior to the departure. We apply the learned seasonal behaviour to the future demand:

$$\hat{y}_t^{hist} = \frac{b_{t,\tau} + s_x^{hist} * (\hat{b}_t - b_{t,\tau})}{\hat{b}_t} \quad (3)$$

with  $x$  the calendar week corresponding to the future time index  $t$ .

An alternative option is to estimate the seasonal behaviour based on the already measured bookings  $b_\tau$ . We can calculate the currently measured seasonal behaviour with  $y_{t,\tau} = \min(\max(\frac{b_{t,\tau}}{E(b_\tau)}, \Phi_{low}), \Phi_{high})$  and estimate the future behaviour with

$$\hat{y}_t^{curr} = \frac{b_{t,\tau} + \Phi_{corr} * y_{t,\tau} * (\hat{b}_t - b_{t,\tau})}{\hat{b}_t} \quad (4)$$

Parameter  $\Phi_{corr} \in [0, 1]$  describes how much we transfer the measured season to the future and how much we apply a ‘no season assumption’ for the future.

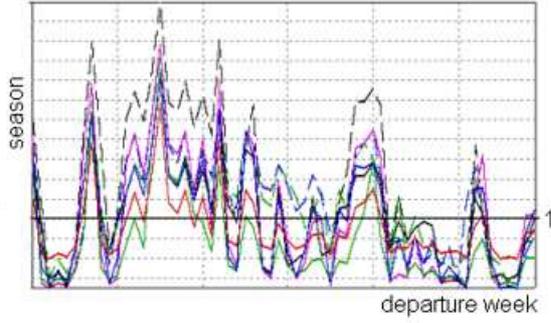


Fig. 6. Subset of diversified predictions for the seasonal behaviour 30 days prior to departure.

### B. Used Diversifications

We have diversified the calculation concerning the three already mentioned criteria: (a) the used method of prediction  $\hat{y}_t^{hist}$  or  $\hat{y}_t^{curr}$ , (b) thick modelling concerning the parameters  $\Phi_I$ ,  $\Phi_{low}$ ,  $\Phi_{high}$  and  $\Phi_{corr}$  and (c) the level of determination of the seasonal factors  $s_x^{hist}$  and the current season  $y_{t,\tau}$  (aggregated over Fareclass and/or Point of Sale and/or Day of the Week). Figure 6 shows an example of a subset of input predictions.

### C. Experimental Results

Table I shows the resulting percentage improvement compared to the current commercial forecasting system using seasonal single level predictions that are already well tuned and optimised. In this system different methods are combined using a predefined rule based system which determines linear combination weights based on different impacts like the size of the numbers to predict. It can be seen that evolving the order of dimensions allows the generation of structures which are only slightly worse than the best found order of dimensions. The evolutionary approach can therefore be evaluated as useful in order to determine the order of dimensions automatically. It can also be clearly seen, that approaches, which allow only the combination of forecasts that have been diversified only by one type of diversification, generally perform better than approaches that do not contain this restriction. But we can also see that there is the risk of instabilities based on overfitting if the potential structures are too flexible. Each of the extensions to the algorithm that we have proposed perform very well, but if the two extensions are applied both, the structures became too flexible and the out of sample forecast accuracy significantly decreases.

## V. SUMMARY AND CONCLUSIONS

We have presented experimental results that support our theoretical findings described in [2]. We have described different evolutionary approaches in order to solve the problem of determining the order of the dimensions used for pooling in our pooling algorithm. Based on an analysis of the resulting structures we have proposed two extensions of such evolutions that allow the generation of more flexible

TABLE I

EXPERIMENTAL RESULTS, THE FIRST COLUMN INDICATES THE STRUCTURE OF THE COMBINATION PROCESS. THEN THE RESULTS ARE REPRESENTED AS AN IMPROVEMENT PERCENTAGE MEASURED ON THE HIGH LEVEL (O&D) AND THE LOW LEVEL (ODIFPOS).

\*:MSMLP STANDS FOR 'MULTI STEP MULTI LEVEL POOLING'

\*\* :FC STANDS FOR 'FORECAST'

structure	high	low
flat comb. (best subset of 5 input fc** at low level)	-5	-3
flat comb. (best subset of 5 input fc** at high level)	+3	+2
flat multi level comb. (best subset of 5 input fc)	+5	+3
flat multi level comb. (best subset of 20 input fc)	-5	-3
static (best found structure)	+9	+7
dynamic evolution [6]	+6	+8
pooling of Aiolfi and Timmermann	+5	+6
msmlp* (best found order of dimensions)	+11	+8
msmlp* (evolved)	+8	+7
msmlp* (evolved + extension 1)	+15	+12
msmlp* (evolved + extension 2)	+11	+11
msmlp* (evolved + both extensions)	+3	+5

multi level multi step combination structures. We compared the results of the algorithm of Aiolfi and Timmermann with flat combination and different combination structures that we have evolved for the application of Revenue Management seasonal forecasting. We could clearly beat the algorithm of Aiolfi and Timmermann and achieve an improvement up to 12 percent compared to the current system.

## REFERENCES

- [1] Riedel, S. and Gabrys, B.(2005). *Combination of Multi Level Forecasts*. Accepted to Special Issue related to Data Fusion, Journal of VLSI Signal Processing Systems, available at <http://dec.bournemouth.ac.uk/staff/bgabrys/>
- [2] Riedel, S. and Gabrys, B.(2006). *Pooling for Combination of Multi Level Forecasts*. available at <http://dec.bournemouth.ac.uk/staff/bgabrys/>
- [3] Granger, C.W.J. and Jeon, Y.(2004). *Thick modeling*. *Econometric Modeling* 21, 323-343
- [4] C.W.J. Granger and Ramanathan,R.(1984). *Improved methods of forecasting*. *Journal of Forecasting* 3, 197-204
- [5] Aiolfi, M. and Timmermann, A.G.(2006). *Persistence of forecasting performance and conditional combination strategies*. *Journal of Econometrics*, vol. 135, issues 1-2, 31-53
- [6] Riedel, S. and Gabrys, B.(2005). *Evolving Multilevel Forecast Combination Models - An Experimental Study*. Proceedings of the NISIS conference 2005, Albufeira, Portugal
- [7] McGill and van Ryzin (1999). *Revenue Management: Research Overview and Prospects*. *Transportation Science* 33, 4
- [8] Riedel, S. and Gabrys, B.(2003). *Adaptive Mechanisms in an Airline Ticket Demand Forecasting System*. Proceedings of the EUNITE'2003 conference, Oulu, Finland
- [9] Ruta,D. and Gabrys, B.(2005). *Classifier Selection for Majority Voting*. *Journal of information fusion, special issue on Diversity in Multiple Classifier Systems*, vol. 6, issue 1, 63-81,
- [10] Riedel, S. and Gabrys, B.(2004). *Hierarchical Multilevel Approaches of Forecast Combination*. Proceedings of the OR'2004 conference, Netherlands