

# Combining regression models and metaheuristics to optimize space allocation in the retail industry

Fábio Pinto<sup>a,\*</sup>, Carlos Soares<sup>b</sup> and Pavel Brazdil<sup>a</sup>

<sup>a</sup>*LIAAD – INESC TEC/FEP, Universidade do Porto, Porto, Portugal*

<sup>b</sup>*CESE – INESC TEC/FEUP, Universidade do Porto, Porto, Portugal*

**Abstract.** Data Mining (DM) researchers often focus on the development and testing of models for a single decision (e.g., direct mailing, churn detection, etc.). In practice, however, multiple decisions have often to be made simultaneously which are not independent and the best global solution is often not the combination of the best individual solutions. This problem can be addressed by searching for the overall best solution by using optimization methods based on the predictions made by the DM models. We describe one case study where this approach was used to optimize the layout of a retail store in order to maximize predicted sales. A metaheuristic is used to search different hypothesis of space allocations for multiple product categories, guided by the predictions made by regression models that estimate the sales for each category based on the assigned space. We test three metaheuristics and three regression algorithms on this task. Results show that the Particle Swarm Optimization method guided by the models obtained with Random Forests and Support Vector Machines models obtain good results. We also provide insights about the relationship between the correctness of the regression models and the metaheuristics performance.

**Keywords:** Data Mining, metaheuristics, retail, space allocation

## 1. Introduction

Most research in Data Mining (DM) is often restricted to a single decision, such as predicting whether a given production plan will yield a faulty product, which customers to target in a marketing campaign or the shelf space to be assigned to a product to optimize its sales. However, in practice, multiple decisions have often to be made at the same time [2]. For instance, in marketing, one might be interested in maximizing the revenue of a set of campaigns and, in retail, the goal may be to maximize the sales over all the products. Moreover, these decisions are often not independent. Taking into account that the number of selected customers for each campaign is limited and that each customer may be contacted at most a given number of times, the selection of customers for one campaign cannot be made without taking into account the others. Similarly, in retail, the total amount of shelf space available in a store is limited and the range of acceptable shelf space for a given product category is subject to business rules. Therefore, an opportunity emerges to develop suitable decision support systems for this purpose that would benefit from approaches combining DM and optimization techniques.

---

\*Corresponding author: Fábio Pinto, LIAAD – INESC TEC/FEP, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal. E-mail: fhpinto@inesctec.pt.

In DM research, the synergies between DM and optimization approaches typically focus on improving the results on a single DM task, such as the tuning of various parameters of a DM algorithm [23] or the search for the optimal set of features for a given task [34]. Thus, although the combination of these two families of techniques is important for a wide range of DM problems [25], it is usually restricted to sub-problems like the ones mentioned above.

One of the industries that can benefit the most from the combination of DM models with optimization techniques is retail. The large amounts of data collected together with the magnitude of the business (many retail chains have hundreds or even thousands of stores) give rise to several opportunities for researchers in DM/optimization.

For retailers, space is one of the most expensive resources [5]. Therefore, one of the most difficult decisions is product category space allocation, and, thus, the definition of a store's layout.<sup>1</sup> When distributing space, it is essential for the retailer to assess the effects of allocating more space to product category 1 instead of product category 2. This problem becomes increasingly complex as the number of product categories increases: the search space is very large.

This paper extends previous work [27] in which we presented a Decision Support System (DSS) to assist retailers in allocating space to a wide range of product categories. The system combined DM models with an evolutionary algorithm and was tested in a case study of a portuguese retail chain. Results showed that our system was able to provide interesting and valuable space recommendations.

In this paper we provide a more detailed description of the proposed framework, the mathematical formalization of the problem, a through review of related work and additional experiments regarding the optimization component and a deeper understanding of the results. The paper is organized as follows. Section 2 presents related work both in retail applications and general DM/Optimization applications. In Section 3, we present our framework for the space allocation problem. Section 4 specifies the data used for this project. In Section 5, we detail the methodology followed for developing our DSS. This includes both the DM and the optimization components. Section 6 shows the experiments carried out and the testing of the system in a case study. Finally, the paper finishes in Section 7 with conclusions and future work.

## 2. Related work

Several DM algorithms include optimization algorithms in such processes, i.e., neural networks, SVM, etc. It is also frequent to find applications of optimization algorithms in tasks such as parameter tuning of a DM algorithm [23] or feature selection [34]. Similarly and in the scope of clustering tasks, van der Merwe and Engelbrecht [32] used PSO to find the centroids of a user specified number of clusters. Cui et al. [7] proposed an hybrid PSO algorithm for document clustering.

The integration of DM models with optimization algorithms has been reasonably explored in the literature. Kusiak and Zheng [21] applied the approach to the optimization of power factor and power output of wind turbines using neural networks and an evolutionary algorithm. Wu and Chen [35] proposed a novel nonparametric regression ensemble rainfall forecasting model integrating generalized Particle Swarm Optimization (PSO) with a neural network. PSO is used to evolve the architecture of the neural network and its connection weights.

---

<sup>1</sup>The store layout is a sketch showing the physical distribution, sizes and weights of elements such as text, graphics or pictures in a store.

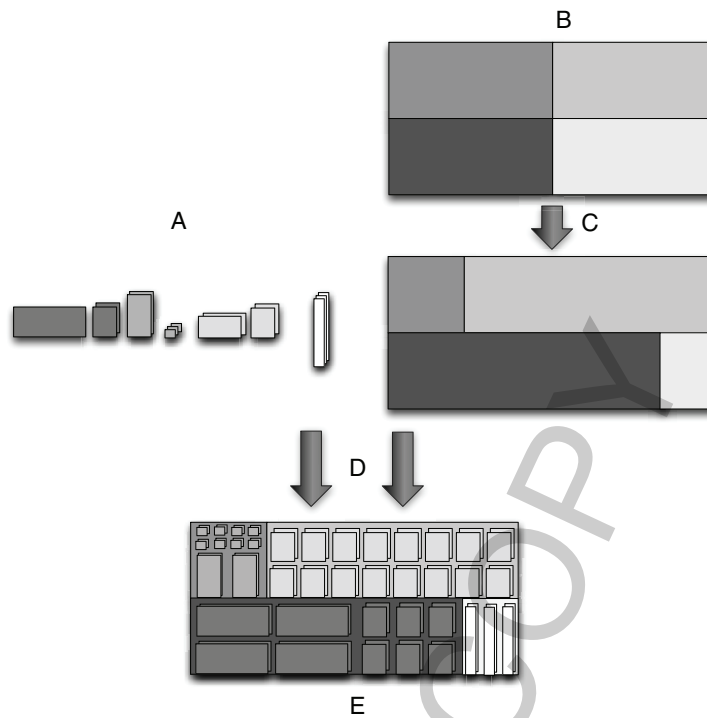


Fig. 1. *Product allocation*. The retailer needs to allocate the products (A) to the available shelf space (B). Each product category has its specific region in the shelf in which its products are going to be allocated. In our work, we optimize the space assigned to each product category, resulting in (C). Then, the products are finally assigned (D) to the shelf optimized space (E). Most of the papers on this subject focus on the latter task. We do not address it.

Regarding our application domain, product space allocation within a retail store is a common research topic. Several studies were conducted to analyze the process of product allocation using econometrics approaches [12], operations research methods [3,6,9,22,36] and even machine learning algorithms. The latter include the application of association rules [24] and genetic algorithms [15]. However, these papers are not concerned with product category space allocation. They focus on distributing products on shelf space. Some of these papers take into account shelf space elasticity<sup>2</sup> and product groupings for determining the distribution of products [3,22]. However, the product category space is determined endogenously along with distribution. In this scenario, the problem is how to accommodate several products into a restricted amount of shelf space. The goal is to select the products that maximize sales while still fulfilling some optimization constraints that are imposed by the business specialists. One can add more variables to this optimization problem if we also consider the number of facings of each product, shelf height, etc. Considering Fig. 1 as an illustrative example, these papers focus on task D.

However, our problem is slightly different. We focus on the problem of deciding how much space in the store should be assigned to each of the product categories, as illustrated in Fig. 1. Posteriorly, a method can be used to define which products are going to be placed in the shelf, as the ones described in the papers mentioned above, but we do not address that problem in this paper.

Desmet and Renaudin [8] published the first paper concerning the problem of product category space

<sup>2</sup>The impact in sales of increasing 1% of shelf space.

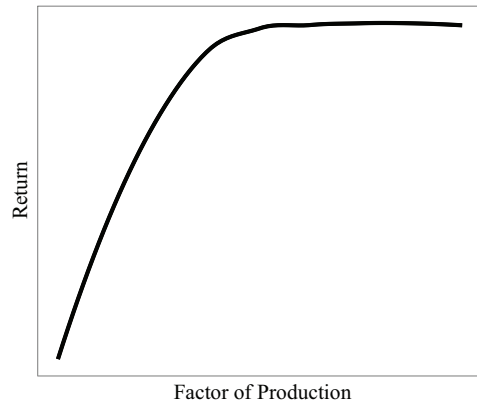


Fig. 2. Law of diminishing returns [30].

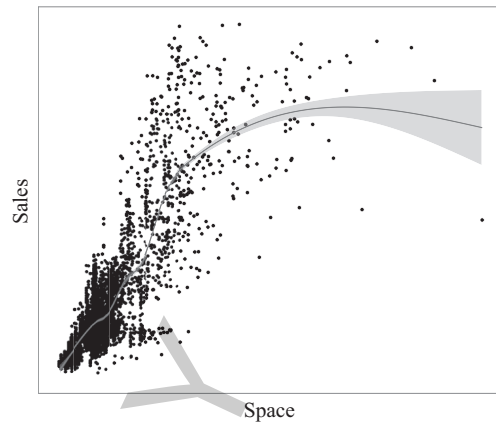


Fig. 3. Sales and space of a product category in our dataset.

allocation. They used econometrics methods to model the sales of product categories and estimated the respective space elasticity. Given the nature of the models generated (linear regressions), there was no purpose in applying an optimization technique. Since linear models capture a linear relationship between sales and space, the product category with greater space elasticity would overpower the optimization process: all the available space would be assigned to that product category. Furthermore, their results were partially questionable, given that some of the estimated space elasticities were negative. In other words, the sales would decrease with the increase in shelf space allocated. Castro [5] followed a very similar approach. Again, the linear nature of the models withdrawn the purpose of an optimization approach. Space elasticities were estimated for each product category. Some of the results in that paper have contributed to the project described in this paper, particularly, the variables used to generate the models.

Recently, Eisend [11] published a meta-analysis study of 1,268 estimates of shelf space elasticities. This work produced several new empirical findings about shelf space elasticity: 1) it was observed that the mean shelf space elasticity is 0.17 and that it varies across product categories, with elasticities increasing as one moves from commodities to staples products and also to impulse purchases; 2) store size moderates the effect of product characteristics on shelf space elasticity. In large stores, the difference between elasticities at the brand and category levels is greater than in small stores. However, the paper does not investigate if the relationship between space and sales of a product category is constant or not. This is a very important issue that is addressed by our approach.

### 3. Proposed framework

The interaction with the business specialists that were involved in this project led to the identification of key issues that are intrinsic to the domain. One of the most important pieces of knowledge obtained is that the relationship between space and sales of a product category is not linear. That is, the effect of increasing space from 10 m<sup>2</sup> to 20 m<sup>2</sup> on the sales of a product category is not the same as the effect of increasing it from 100 m<sup>2</sup> to 110 m<sup>2</sup>. This is known in Economics as the Law of Diminishing Returns [30]: adding one more unit of a factor of production, while holding all others constant, will at some point yield lower per-unit returns. This concept is illustrated in Fig. 2. Figure 3 presents the sales

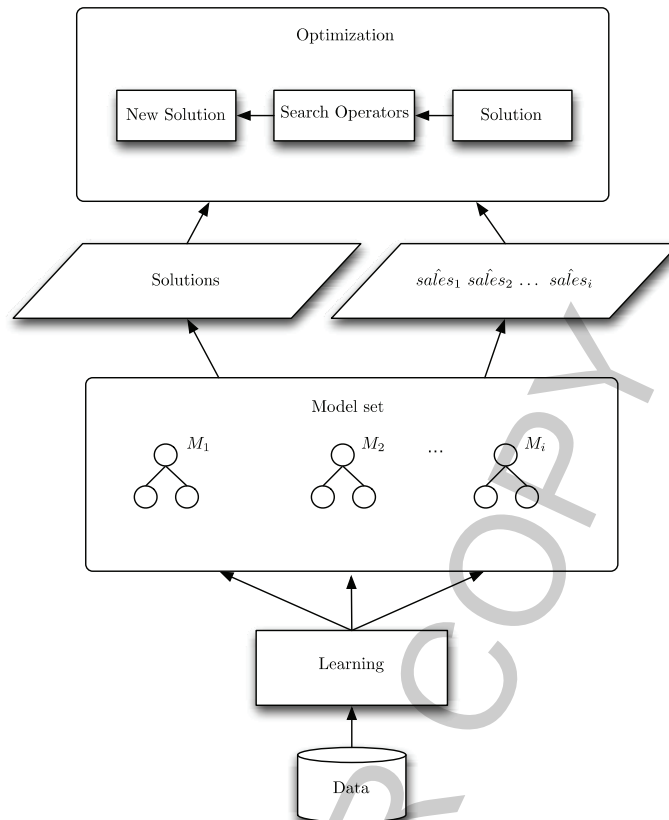


Fig. 4. Proposed framework of a DSS for space allocation.

of a product category for different shelf space allocations. We can see empirically that the relationship between sales and space follows that law.

It is obvious that the models of shelf space elasticity must take this into account. However, as mentioned previously, the solutions proposed in the literature for this problem fail to do so [5,8]. Linear models (like linear regression) cannot capture relationships such as the one presented in Fig. 3. Our approach takes this issue into account by applying regression algorithms that are able to capture non-linear relationships in the data.

The framework used in this project is illustrated in Fig. 4. It starts by generating sales forecasting models for each product category, based on historical data. Given that the sales of a product category are not only dependent on the allocated shelf space, other relevant variables must be incorporated into the models. Thus, these models estimate sales based on a group of variables, including space. This is the DM component of our Decision Support System (DSS). We carefully selected regression algorithms that we knew a priori that could capture the non-linear relationship between sales and space.

Then, the models are used to estimate the sales for every product category according to the space distribution that is under evaluation. These estimates are used by a metaheuristic optimization method to guide the search for the solution that maximizes the overall sales. The metaheuristic iteratively generates solutions, i.e., space allocation for all product categories, and the models predict the sales for every category. These predictions are finally added up to quantify the quality of the solution under evaluation.

The optimization process has two main hard constraints: a minimum and maximum value of space for each product category; and the sum of all values can not be larger than the total space of the store.

We mathematically formalize the problem here as

$$\begin{aligned} &\text{maximize } S = \sum_i \hat{\text{sales}}(a_i) \\ &\text{subject to } \sum_i a_i = \bar{A}, \text{ } llimit_i \leq a_i \leq ulimit_i \end{aligned} \quad (1)$$

in which  $S$  is the total sales of a store,  $\hat{\text{sales}}$  and  $a$  are the sales and area assigned to a product category  $i$ , respectively,  $\bar{A}$  is the total area of the store,  $llimit_i$  is the lower limit of space assigned to  $i$  and  $ulimit_i$  is the upper limit of space assigned to  $i$ .

The sales forecasting models to predict  $\hat{\text{sales}}_i$  are going to be developed using variables describing the area, the stores, the product categories and seasonality:

$$\hat{\text{sales}}_i : \mathbb{N}^{\text{area}} \times D^{\text{store}} \times D^{\text{category}} \times \mathbb{N}^{\text{seasonality}} \rightarrow \mathbb{R}^{\text{sales}} \quad (2)$$

#### 4. Data

This section specifies the data collected for the project. The dataset comprises two years of data with monthly observations for 110 product categories. Overall, it contains 332,885 observations.

The majority of the variables were provided by the retail company in which this project was developed. Due to confidentiality reasons, we can not give insights about their construction but we can motivate the purpose for their inclusion in our dataset. For all variables,  $i$  represents a product category,  $m$  represents a month and  $s$  a store.

- **sales<sub>*i,m,s*</sub>**. The target variable.
- **a<sub>*i,m,s*</sub>**. Total area,<sup>3</sup> in square meters, assigned to a product category. [5] showed its significance on sales forecasting models for retail.
- **m**. Month of the example. Included for seasonality purposes. It is common sense that retail sales are highly seasonal and we expect to model that volatility with this nominal variable.
- **Insignia<sub>*s*</sub>**. Insignia of the store. The retail company has three different types (insignias) of stores. This nominal variable captures different sales behaviour among these insignias.
- **Cluster<sub>*s*</sub>**. Sales potential cluster of the store (nominal variable). The retail company divides its stores into four distinct clusters according to their sales potential.
- **Cluster\_Client<sub>*s*</sub>**. Client profile cluster of the store. The retail company divides its stores in four distinct clusters according to the profile of their customers. Again, the inclusion of this nominal variable seemed relevant given that it is expected that different customers will result in stores with different sales behavior.
- **PPI\_County<sub>*s*</sub>**. Purchasing Power Index of the region in which the store is located. It is expected that the larger the value of this variable, the larger the value of sales.

---

<sup>3</sup>Sum of permanent and promotional area.

- **N\_W\_Days<sub>m</sub>**. Number of non-working days of the month. Customers do most of their shopping on non-working days so it is expected that the larger the value of this variable, the larger the values of sales.
- **C\_P\_Index<sub>i,s</sub>**. Category penetration index by store's client profile cluster. This is a discrete variable calculated for each product category within each customer cluster, so, there are 4 indexes by product category, one for each cluster. This variable can capture the impact that different types of customers have in product category sales.

Other variables, used in [8], contain information that is expected to be useful for the problem. However, this is the data that was available for the project. According to the domain experts, these variables contained sufficient information for the task.

## 5. Methodology

This section specifies the methodology used for developing the predictive models and integrating them with a metaheuristic. The DM problem is addressed as a regression task with  $sales_{i,m,s}$  as the dependent variable. We describe how error is estimated and which regression algorithms were selected for the task. We also give insights about the metaheuristics tested as the optimization component of our DSS. References are provided that contain more detailed information about all the methods used.

### 5.1. Error estimation

The error measures for the evaluation of the predictive models are: *Mean Relative Error (MRE)*, defined as

$$\frac{\sum_{j=1}^n \left| \frac{y_j - \hat{y}_j}{y_j} \right|}{n} \times 100$$

*Root Mean Squared Error (RMSE)*, defined as

$$\sqrt{\frac{\sum_{j=1}^n (y_j - \hat{y}_j)^2}{n}}$$

and *Variation Index (varIndex)*, defined as

$$\frac{RMSE}{\bar{y}}$$

For all measures,  $y_j$  is the true value,  $\hat{y}_j$  is the predicted value, and  $\bar{y}$  is the mean of the output variable.

The performance of the regression algorithms will be compared with a baseline, whose predictions consist of the average value of the target variable for the store which is being analyzed. This comparison will help assess the difficulty of the phenomenon that we are modeling as well as how much useful knowledge the regression models are capturing. The retail company defined 10% as the (maximum) target MRE value. We used this value as a threshold for success of our models.

Given that the dataset used in this work consisted of a time series, the error was estimated using a hold-out strategy, ensuring that the test data was more recent than training data. The training set for each product category (and each model) consisted of one year and four months of observations; the remaining examples (eight months) were splitted (four months each) for a validation and a test sets. The validation set was used for parameter tuning and the test set to assess the generalization error of the models.

### 5.2. Regression algorithms

In previous work [27] we showed results of a preliminary experiment in which we tested 6 regression algorithms. Those results indicate that the algorithms that are more suited for this task are:

- *Support Vector Machines* (SVM) with Gaussian radial basis function [33];
- *Support Vector Machines* (SVM) with sigmoid function;
- *Random Forests* (RF) [4].

The implementation of these algorithms available in the *R* [29] software was used in the experiments.

### 5.3. Metaheuristics

Previously, we developed an adaptation of the Genetic Algorithm (GA) for our optimization problem. The algorithm is fully described in [27]. Now, we extend that work by testing two more metaheuristics. For these, we used the packages *GenSA* (for the implementation of the Simulated Annealing algorithm) and *pso* (for the PSO algorithm) of the *R* [29] framework.

#### *Genetic algorithm*

Proposed by Holland [14], this algorithm is considered the root of Evolutionary Computation [10], a framework for population-based search and optimization that is inspired by the evolutionary theory. The main features of our implementation to the standard GA are: crossover with whole arithmetic recombination; mutation of two genes of a solution by adding and subtracting the value 1; and an elitist survival selection. This last feature was implemented given the very small range of fitness values of the solutions in the population: if the survival selection was probabilistic, good solutions were often discarded when building a new generation.

#### *Simulated annealing*

Kirkpatrick et al. [18] developed the Simulated Annealing (SA) algorithm for global optimization. This algorithm is inspired in the process of physical annealing with solids, in which a crystalline solid is heated and then allowed to cool very slowly until it achieves its most regular possible crystal lattice configuration and, thus, is free of crystal defects. If the cooling schedule is sufficiently slow, the final configuration results in a solid with such superior structural integrity. Simulated annealing establishes the connection between this type of thermo dynamic behavior and the search for global minima in a discrete optimization problem [19].

#### *Particle swarm optimization*

PSO has its roots in the work of Kennedy and Eberhart [16]. This evolutionary computation technique was based on the movement of swarms while looking for the most fertile feeding location. By changing the velocity of each particle, the algorithm evolves for a better solution. For more details, we refer the reader for other literature regarding swarm intelligence [17].

## 6. Experiments

This section presents the experiments carried out with each component of our DSS. We firstly describe the process of generating and evaluating the sales forecasting models. Based on these results, we selected the best generated model for each product category. Next, we evaluate the performance of the three selected metaheuristics.



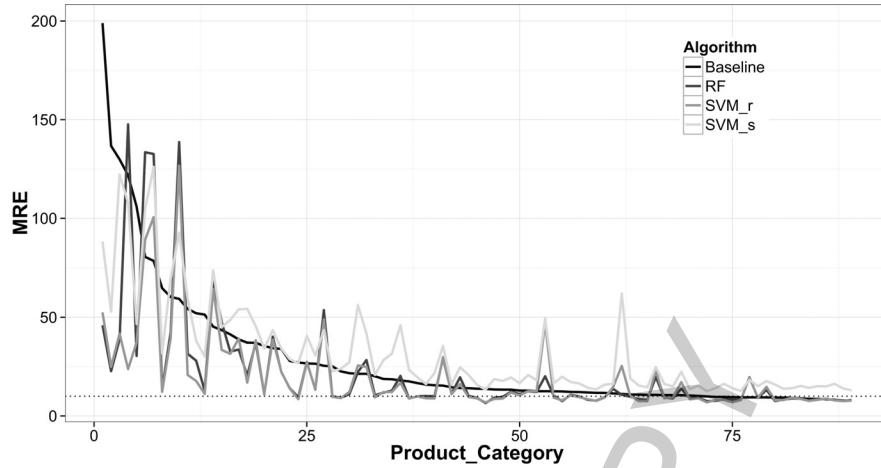


Fig. 5. Results of the three regression algorithms in terms of MRE.

## 6.1. Sales forecasting models

### 6.1.1. Experimental setup

Given that some of the 110 product categories of our dataset showed a very low number of observations, we only used the 89 categories with more than 1000 observations. Additionally, for 10 of these product categories, we only had one year of data. Therefore, for this group of product categories we used 6 months for training; 3 months for validation and the remaining 3 months for testing.

The parameter optimization of the algorithms was carried using the *R* package *caret* [20].

### 6.1.2. Results

Figure 5 shows the results obtained for the three regression algorithms tested in terms of MRE on 89 product categories. The dotted horizontal line represents the threshold defined in Section 5. We can see that several models are below or very close to the threshold. More particularly, 42% of the models are below the 10% threshold in terms of MRE and 66% are below 15%.

We verified the statistical significance of the error differences of the models using paired t-tests with  $\alpha = 0.05$ . This process was done for all the error measures specified in Section 5. Overall, SVM with the radial kernel obtained the best performance in 77 product categories; RF obtained the best performance in 10 product categories; and finally, SVM with sigmoid kernel presented superior performance in only two product categories. In several product categories, RF and SVM with radial kernel function showed very similar performance levels and the statistical test was not able to distinguish between them. In these cases, we chose SVM because it is computationally less demanding than RF.

## 6.2. Optimization

The case study used in this experiment concerns the optimization of the space assigned to 85 product categories within a retail store. Given the makeover of a store of the retail company, we compared the recommendations made by the system with those of the business specialists. Previous results showed that the final solution of the system presented a Pearson's correlation of 0.66 to the solution found by the business specialists [27]. This result gave confidence about the quality of the work that was being developed. Now, we extend that work by testing different metaheuristics as optimization components.

### 6.2.1. Experimental setup

Given that the variable to be optimized is a numeric variable, the problem turns into an optimization of 85 numeric values. For each one of the numeric values there is a minimum and a maximum value, defined by the business specialists of the retail company. These values are very diverse: for some product categories it is between 5 and 10 m<sup>2</sup> but several have a larger range, i.e., 25–450 m<sup>2</sup>. These range constraints are provided to all the metaheuristics, to ensure that the solutions generated by the methods are valid.

The fitness function is the same for all metaheuristics and includes a penalty factor  $p$  that penalizes a solution if the sum of all its values is not within a specific interval: this represents the constraint of the total space of the store.

$$Fitness_k = sales_{1,m,s} + sales_{2,m,s} + \dots + sales_{i,m,s} + p$$

The interval was set by the business specialists. This is a hard constraint. A solution is useless if it does not satisfy this constraint. Therefore, we set the penalty value as 1/3 of the fitness obtained by the initial solution. This initial solution is provided by the business specialists and satisfies the total space constraint. Preliminary experiments in which an initial solution was not provided showed that none of the metaheuristics that we implemented could achieve a solution that satisfied the total space constraint. This is a clear indicator of the very large search space of this optimization problem and the importance of incorporating domain knowledge into the methods that are integrated into the DSS.

Preliminary experiments showed that PSO required more parameter tuning than GA and SA. PSO showed high sensitivity to parameter changes. Our approach to tune its parameters was based on two existing methods [26,31]. The parameters of the PSO algorithm were:  $s = 140$  (swarm size),  $w = -0.2$  (exploitation constant),  $c.p = 0.1$  (local exploration) and  $c.g = 3.75$  (global exploration).

Previous results showed that GA stabilized after 60,000 iterations [27]. In this work, we wanted to test if by increasing the number of iterations we could retrieve better solutions. With this in mind, we set the number of total iterations to 75,000 for all experiments.

The results presented in this paper have been transformed due to confidentiality reasons: all fitness values were divided by the value  $F$ , that represents the maximum fitness value that we could retrieve from a solution in all experiments that we conducted. Therefore, all fitness values are scaled between 0 and 1.

### 6.2.2. Results

Figure 6 presents the performance of the three metaheuristics with 75,000 iterations. PSO clearly shows a better performance than the other two: it achieves solutions with higher fitness<sup>4</sup> and it does so in significantly fewer iterations than GA and SA. These results indicate also that both GA and SA converge to a local maximum. However, there is also no guarantee that the result obtained by PSO is not, also, a local maximum.

Figure 7 presents a different perspective on the performance of the three metaheuristics. The x-axis shows the number of evaluations made by the fitness function for each algorithm. In other words, it measures the number of different solutions that were tested. Again, PSO requires the least number of evaluations to stabilize and presents the best solution. On the other hand, GA shows much better performance than the SA from this evaluation perspective.

The graph in Fig. 8 refers to the same set of experiments. However, the  $x$ -axis represents time, mea-

<sup>4</sup>The solution obtained by PSO presents the highest fitness from all the solutions found in our experiments. Therefore, we normalized all the results according to this value.

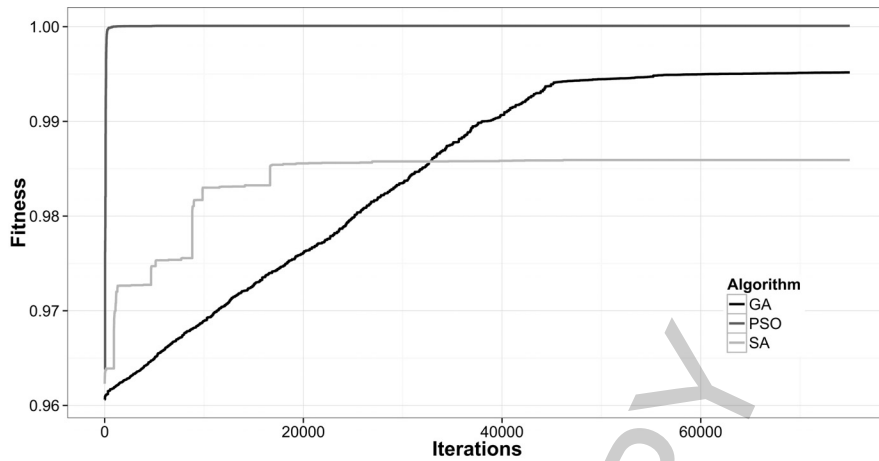


Fig. 6. Results of the three metaheuristics in terms of number of iterations.

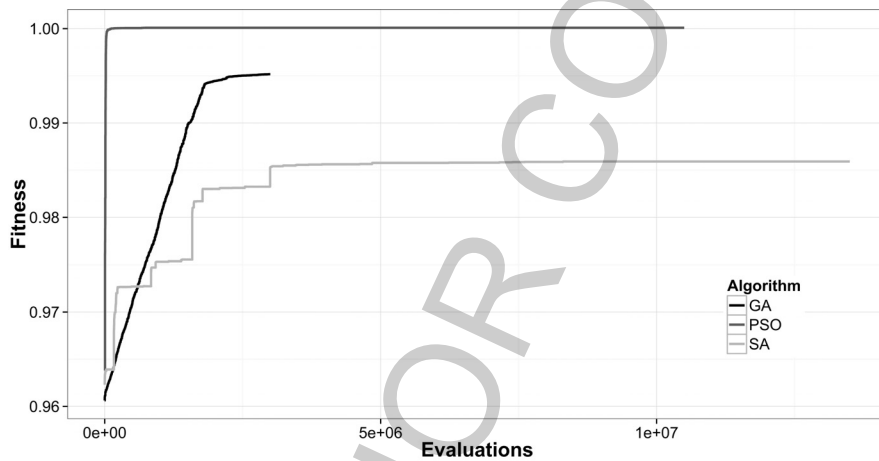


Fig. 7. Results of the three metaheuristics in terms of number of evaluations.

sured in minutes, instead of iterations. As mentioned previously, GA needs a lot of time to stabilize in a solution and this graph shows it clearly. It is by far the slowest metaheuristic. On the other hand, SA is faster but stabilizes in a solution that is inferior to the solution found by GA. According to the business specialists, time is an important issue. However, the ability to achieve better solutions is more important. Additionally, it may be the case that the implementation of GA, which was made by the authors of this work, is less efficient than the implementations of the other metaheuristics, which are available in *R*.

Comparing PSO with the other metaheuristics in terms of time, again we observe that this metaheuristic presents a far superior performance. Only a few minutes are enough to achieve a solution very similar to  $F$ , the maximum fitness value observed.

Despite superior results, PSO also has disadvantages. It is the most unstable metaheuristic tested. Given the randomness components of the algorithm, the performance that we see in the previous graphs is not always true. More importantly, a fine tuning of the algorithm is essential to obtain good results. In fact, preliminary experiments showed that without that tuning, its performance was only slightly better than that of SA.

Table 1

Results of the experiments using neural networks (NN) as regression algorithm in comparison with RF/SVM. *Corr* corresponds to the Pearson's correlation with the space distribution defined by the business specialists and *%Improvement* to the sales gain of the optimized distribution predicted by the models

Algorithm	<i>Corr</i>		<i>%Improvement</i>	
	RF & SVM	NN	RF & SVM	NN
GA	0.7387	0.7401	6.32%	6.31%
PSO	0.6909	0.6788	6.41%	6.43%
SA	0.7546	0.7495	5.72%	6.20%

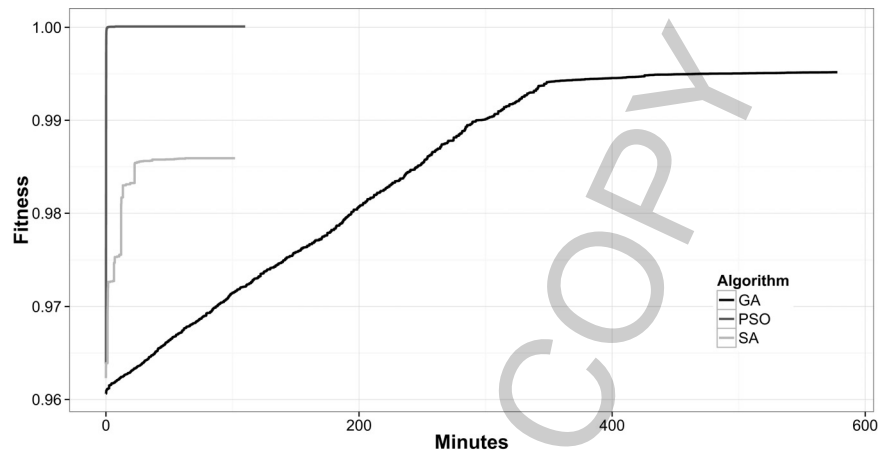


Fig. 8. Results of the three metaheuristics in terms of time, measured in minutes.

### 6.3. Discussion

The results obtained both for the sales forecasting models and optimization experiments are satisfactory. However, it is important to state that the success of the DSS is essentially linked with the correctness of the regression models. The key to achieve good layout recommendations is the ability of the models to describe the relationship between the space allocated to a product category and its respective sales. We executed some experiments to further investigate this issue.

As showed in [27], neural networks do not perform particularly well on this problem. Although it is not the regression algorithm with worst performance, SVM and RF show unequivocal better performance. Therefore, we executed experiments (see Table 1) in which we used neural networks as regression algorithm for all product categories with the three tested metaheuristics as optimization component. Our goal with these experiments is to gain a better understanding of the relationship between the correctness of the regression models and the optimization metaheuristics.

Results show that the space distribution suggested by the DSS with neural networks or with a combination of RF/SVM is very similar. The fact that different regression algorithms capture a similar relationship between sales and space is an encouraging result. On the other hand, it raises questions on how this kind of DSS should be evaluated. It is really important to evaluate the models individually or should we focus on evaluating the optimization procedure and its impact in terms of sales? The latter approach resembles utility based data mining (UBDM) [28] research which takes into account economic utility factors (such as costs) to evaluate data mining systems. We believe that our work could benefit from such approach but further research is needed to assess that hypothesis.

Another important discussion is concerned with the generalization ability of the framework that we

propose in this paper. We would like to investigate if the combination of predictive models and metaheuristics as it was used in this paper can be effective in other applications [13].

## 7. Conclusions and future work

In this paper, we provide an empirical study of a framework that combines DM models with an optimization technique to recommend space distributions of product categories within a retail store. As far as we know, our approach is the first to combine machine learning and optimization techniques to deal with this problem.

Our approach has two components: the sales forecasting models and the optimization metaheuristic. For the first component, we generated and empirically tested predictive models to estimate the sales for each product category based on the assigned space and other variables. Our models are generated using regression algorithms that are able to capture the non linear nature of the relationship between sales and space. This is a very important point that distinguishes our work from previous approaches.

For the second component, the predictive models are used to guide the search of a metaheuristic for a solution that maximizes the overall sales. The metaheuristic generates solutions containing the space allocation for all categories and the models make predictions concerning the sales for each category. The optimization component makes it possible to take into account the non linear impact of other product categories in the layout of the retail store. From the three metaheuristics tested, Particle Swarm Optimization showed the best results, both in terms of the quality of the solution obtained and the computational cost.

The testing of the DSS in a case study showed that it is able to provide space recommendations considered useful and interesting by the business specialists.

As for future work, for some product categories, the number of examples was small. This problem can be resolved by collecting more data. Ideally, this new data would reflect a systematic variation of product category space within a store(s), allowing for a more representative sample of the phenomenon. To reduce the impact of this process on the store and, thus, on the customers, active learning and cost-sensitive approaches could be investigated.

We have developed models that are independent of each other. However, it is likely that there is some dependence between the sales of at least some of the product categories. We plan to test multi-target regression methods [1] to address this problem.

We are also interested in further investigating new evaluation frameworks for our problem and similar ones, more particularly, frameworks that take into account economic utility factors.

## Acknowledgements

This work is partially funded by FCT/MEC through PIDDAC and ERDF/ON2 within project NORTE-07-0124-FEDER-000059 and NORTE-07-0124-FEDER-000057. The research leading to these results has also received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° [260169] (Project CoReNET). The authors would also like to thank the retail company for providing the data and support during the project.

## References

- [1] T. Aho, B. Zenko, S. Dzeroski and T. Elomaa, Multi-target regression with rule ensembles, *Journal of Machine Learning Research* **1** (2012), 1–48.

- [2] M.J. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*, John Wiley & Sons, Inc., 1997.
- [3] M. Bilsel, M.B. Ayhan and S. Bulkan, Shelf space optimization using metaheuristic algorithms, *Business and Management* **5**(2) (2013).
- [4] L. Breiman, Random forests, *Machine Learning* **45**(1) (2001), 5–32.
- [5] A. Castro, A.O. Brochado and F.V. Martins, Supermarkets sales and retail area: A mixture regression model for segmentation, in: *Poster Presentation in European Network for Business and Industrial Statistics*, (2007).
- [6] M. Corstjens and P. Doyle, A model for optimizing retail space allocations, *Management Science* **27**(7) (1981), 822–833.
- [7] X. Cui, T.E. Potok and P. Palathingal, Document clustering using particle swarm optimization, in: *Swarm Intelligence Symposium*, IEEE (2005), 185–191.
- [8] P. Desmet and V. Renaudin, Estimation of product category sales responsiveness to allocated shelf space, *International Journal of Research in Marketing* (15) (1998), 443–457.
- [9] X. Dréze, S.J. Hoch and M.E. Purk, Shelf management and space elasticity, *Journal of Retailing* **70**(4) (1994), 301–326.
- [10] A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computing*, volume 2, Springer Berlin, 2010.
- [11] M. Eisend, Shelf space elasticity: A meta-analysis, *Journal of Retailing* (2013).
- [12] V. Gaur, M. Fisher and A. Raman, An econometric analysis of inventory turnover performance in retail stores, *Management Science* **51** (2005), 181–193.
- [13] C. Gomes, B. Almada-Lobo, J. Borges and C. Soares, Integrating data mining and optimization techniques on surgery scheduling, in: *Advanced Data Mining and Applications*, Springer (2012), 589–602.
- [14] J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and artificial intelligence*, MIT Press, 1992.
- [15] H. Hwang, B. Choi and G. Lee, A genetic algorithm approach to an integrated problem of shelf space design and item allocation, *Computers and Industrial Engineering* (56) (2009), 809–820.
- [16] J. Kennedy and R. Eberhart, Particle swarm optimization, in: *IEEE International Conference on Neural Networks*, volume 4, IEEE, (1995), 1942–1948.
- [17] J.F. Kennedy, J. Kennedy and R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Pub, 2001.
- [18] S. Kirkpatrick, D. Gelatt, Jr. and M.P. Vecchi, Optimization by simulated annealing, *Science* **220**(4598) (1983), 671–680.
- [19] F. Glover and G.A. Kochenberger, *Handbook of Metaheuristics*, Springer, 2003.
- [20] M. Kuhn, Building predictive models in r using the caret package, *Journal of Statistical Software* **28**(5) (2008), 1–26.
- [21] A. Kusiak and H. Zheng, Optimization of wind turbine energy and power factor with an evolutionary computation algorithm, *Energy* **35**(3) (2010), 1324–1332.
- [22] A. Lim, B. Rodrigues and X. Zhang, Metaheuristics with local search techniques for retail shelf-space optimization, *Management Science* **50**(1) (2004), 117–131.
- [23] S.-W. Lin, K.-C. Ying, S.-C. Chen and Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, *Expert Systems with Applications* **35**(4) (2008), 1817–1824.
- [24] M. Nafari and J. Shahrabi, A temporal data mining approach for shelf-space allocation with consideration of product price, *Expert Systems with Applications* (37) (2010), 4066–4072.
- [25] B. Padmanabhan and A. Tuzhilin, On the use of optimization for data mining: Theoretical interactions and ecrm opportunities, *Management Science* **49**(10) (2003), 1327–1343.
- [26] M.E.H. Pedersen, Good parameters for particle swarm optimization, 2010.
- [27] F. Pinto and C. Soares, Space allocation in the retail industry: a decision support system integrating evolutionary algorithms and regression models, in: *Machine Learning and Knowledge Discovery in Databases*, Springer, (2013), 531–546.
- [28] S. Puuronen and M. Pechenizkiy, Towards the generic framework for utility considerations in data mining research, in: *Data Mining for Business Applications*, IOS Press (2010), 49–65.
- [29] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria (2012).
- [30] P. Samuelson, *Microeconomics*, McGraw-Hill/Irwin, 18th edition, 1948.
- [31] I.C. Trelea, The particle swarm optimization algorithm: Convergence analysis and parameter selection, *Information Processing Letters* **85**(6) (2003), 317–325.
- [32] D.W. Van der Merwe and A.P. Engelbrecht, Data clustering using particle swarm optimization, in: *Congress on Evolutionary Computation*, volume 1, IEEE (2003), 215–220.
- [33] V. Vapnik and C. Cortes, Support-vector networks, *Machine Learning* (20) (1995), 273–297.
- [34] X. Wang, J. Yang, X. Teng, W. Xia and R. Jensen, Feature selection based on rough sets and particle swarm optimization, *Pattern Recognition Letters* **28**(4) (2007), 459–471.
- [35] J. Wu and E. Chen, A novel nonparametric regression ensemble for rainfall forecasting using particle swarm optimization technique coupled with artificial neural network, in: *Advances in Neural Networks*, Springer (2009), 49–58.
- [36] M. Yang, An efficient algorithm to allocate shelf space, *European Journal of Operational Research* **131**(1) (2001), 107–118.