Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA. **Rect@**
Volumen 18 (2017), páginas 71 a 92.
DOI: 10.24309/recta.2017.18.1.05

EXACT METHODS FOR VARIABLE SELECTION IN LINEAR REGRESSION WITH SUB-SETS: ANALYSIS OF DIFFERENT TOOLS AND STRATEGIES

JOAQUÍN ANTONIO PACHECO BONROSTRO
jpacheco@ubu.es
Universidad de Burgos/ Departamento de Economía Aplicada
Plaza Infanta Elena s/n, 09001 Burgos, España

SILVIA CASADO YUSTA
scasado@ubu.es
Universidad de Burgos/ Departamento de Economía Aplicada
Plaza Infanta Elena s/n, 09001 Burgos, España

Recibido (11/09/2017)
Revisado (23/11/2017)
Aceptado (27/11/2017)

RESUMEN: En este trabajo se analiza un problema de selección de variables para regresión lineal. En este caso el conjunto de variables independientes se particiona en grupos disjuntos. El problema consiste en la selección de variables, pero con la restricción consistente en que el conjunto de variables que se seleccione debe de tener al menos una variable de cada grupo. Este problema tiene múltiples aplicaciones, concretamente el diseño de los indicadores sintéticos en diferentes áreas (sociología y economía entre otras). Los diferentes grupos de variables corresponden a los diferentes puntos de vista del problema que se está analizando. Por lo tanto estos indicadores deben de contener variables de todos los grupos. Para resolver este problema se propone un método de Branch & Bound que obtiene soluciones exactas. Además, se proponen y analizan diferentes estrategias para reducir los tiempos de cálculo de este método. Se han realizado diferentes experimentos computacionales que muestran los buenos resultados de ambas estrategias, (tanto por separado como conjuntamente): consiguen reducir notablemente los tiempos de cálculo del método Branch & Bound y permiten resolver problemas de tamaño mayor.

Palabras claves: Selección de variables; Indicadores sintéticos; Método de Branch & Bound; Preselección; Heurísticos.

ABSTRACT: The abstract should summarise the context, contents and conclusions of the paper in less than 200 words preferably in less than 150 words. It should not contain any references or displayed equations. Typeset the abstract in Times New Roman and indent the text. Type similarly the keywords below. A variable selection problem in the context of Linear Regression is analyzed. In this case, the set of original independent variables is partitioned into disjoint groups. The problem consists in the selection of independent variables, but with one restriction: the set of variables that is selected should at least have one variable from each group. This problem has a wide scope of application, specifically the design of composite indicators in different areas (sociology, and economy, among others). The different groups of variables correspond to different viewpoints of the problem under analysis. Therefore, these indicators should contain independent variables from all the groups. For this problem, a Branch & Bound method is proposed to obtain optimal solutions. Moreover, two strategies are proposed and analyzed, to reduce the calculation times of this method. Different computational experiments were completed that showed the good results of both strategies, (both separately and jointly): they managed to reduce the calculation times of the Branch and Bound method considerably, thereby offering solutions to moderated-sized problems.

Keywords: Variable selection; Composite indicators, Branch & Bound methods; Pre-selection, Heuristics.

1. Introduction

1.1. *Motivation*

Research frequently generates datasets comprising one dependent or response variable and multiple independent or predictor variables, giving a dataset that is multivariate and multidimensional. Traditional analysis of such datasets has been based on General Linear Models often using Multiple Linear Regression. Other more recent methods are based on neural networks, support vector machine, nearest neighbor, etc. In the simplest case the Multiple Linear Regression implies a regression of the selected dependent variable with respect to the complete suite of predictor variables. Although this full model regression approach might seem logical, there are several key problems. One of the most important is the following: having multiple predictors in a model adds noise to the analysis, with the effect that non-significant results may be returned, even when the model contains significant predictors (Mundry and Nunn 2009).

In earlier works on the selection of variables, such as those mentioned in sub-section 1.2, there are no prior restrictions on the sets of variables to select. All in all, in some works the maximum size of the sub-set of variables to select should not exceed a previous value. However, to the best of our knowledge, no other type of prior restrictions are considered in the previous literature (except in the work that is cited in the following paragraph).

In this work, a special variable selection problem for linear regression is analyzed. Specifically, the set of original independent variables is partitioned into disjoint groups and the set of variables that is selected should contain elements from all the groups. We consider the following constraint, in order to ensure the "participation" of every group in the final solution: the sub-set of selected variables must include at least one variable from each group. In addition, it ensures that all viewpoints are considered and helps to avoid the selection of variables with high correlations between each other (in general the variables of the same group are usually more highly correlated between each other than with the rest of the group). This constraint has been considered in Pacheco et al. (2013) in the Principal Component Analysis (PCA) context. To our knowledge there are no other references in the literature on this specific variable selection problem (that is, considering this constraint). We go on to explain the importance of this constraint.

As has been commented in Pacheco et al. (2013), in many studies the initial variables are divided into previously selected groups. In these cases it is required, or at least recommended to use variables from all the groups under consideration. This happens, for example, in the construction of composite indicators that are used in several areas (economy, society, quality of life, nature, technology, etc.). The composite indicators are used as measures of the evolution of regions or countries in such areas. The synthetic indicators should try to cover all points of view of the problem (which may be identified with each of the different groups of variables). To do so, they should therefore contain at least one variable from each group (or other types of similar conditions), so that they encompass all the points of view. The importance of composite indicators is explained in Nardo et al (2005a and 2005b) and Bandura (2008), among other references. The convenience of using variables from all groups under consideration is at least explicitly mentioned in Nardo et al. (2005a), Ramajo-Hernández and Márquez-Paniagua (2001) and López-García and Castro-Núñez (2004). There are previous groups of variables and every group participated in the final Composite Indicator, in several of the examples mentioned in the above references and links (and in others). Thus, in Tangian (2007), an example was given of building a composite indicator to measure the working conditions. In this work, 10 groups of variables are considered (Physical environment, Health, Time factors, etc.). In Chan et al (2005), a composite indicator was built to be used as an analytical tool to examine the quality of life in Hong Kong. The Index is now released annually. It consists of 20 variables that are grouped into three groups: Social (10 variables), economic (7) and environmental (3). In Nardo et al. (2005a and 2005b), a Technology Achievement Index was proposed, and the following groups are considered: creation of technology (2 variables); diffusion of recent innovations (2); diffusion of old innovations (2); and, human skills (2). In López-García and Castro-Núñez (2004), an indicator for regional economic activity was constructed for Spain. The following groups are considered: Agriculture (2 variables), Construction (5),

Industry (4), Merchandise Services (9), Non-Merchandise Services (2). In Blancas et al (2010) composite indicators to evaluate tourism sustainability are proposed and Parada et al (2015) obtain a synthetic indicator allowing the measurement of the degree of academic excellence. There are several further examples in the literature, some of them may be found in Bandura, (2008), an annual survey with around 170 international composite indicators.

As explained in the above paragraphs, the importance of this restriction is evident, above all in the studies that wish to reflect the impact of the different aspects or viewpoints of the study (which correspond with the different groups). It is, for example, what happens in the design of composite indicators, as has been described above. So, in this work, an exact method is proposed to search for optimal solutions to this problem with data bases with a moderate number of variables. In addition, different variants and strategies are analyzed, to shorten the computation time. The contributions are described in greater detail in sub-section 1.3.

1.2. *Related Literature*

From a computational point of view, variable selection in regression and for classification is a NP-hard problem (Cotta et al. 2004). So the optimal solution is only found following lengthy computation times expended in large datasets. Therefore, several approximate methods have been proposed. For example, the conventional variable selection strategies involving sequential searches (forward selection, backward elimination, or stepwise selection) by using different goodness-of-fit measures such as the adjusted R2, Akaike Information Criterion (AIC), the Bayesian Information Criteria (BIC) and Mallows Cp. These methods have several well-acknowledged shortcomings: They will not always provide the best subset, they become increasingly ineffective in higher dimensions and show high sensitivity towards small changes in the data (Fan and Li 2001). The stepwise selection procedures are also prone to getting trapped in locally optimal models (Hocking 1976) and face design problems with complex patterns of multicollinearity (Hans and Dobra 2007). Despite the drawbacks, they are still the immediate choice in routine data analysis and because of their simplicity are applied in large data bases (Luo and Ghosal 2016). Least Angle Regression, LARS (Efron et al. 2004) is another method that sequences the candidate predictors in order of importance. Another approach is the addition of a penalty term to the objective function of least squares regression to ensure the sparcity of the model. For example the LASSO method (Least Absolute Shrinkage and Selection Operator; Tibshirani 1996) and Bridge (Frank and Friedman 1993; Fu 1998). Fan and Li (2001) used another penalty function, namely Smoothly Clipped Absolute Deviation (SCAD). Finally the Nonnegative Garrote (Breiman 1995) uses a penalty on shrinkage factors of the regression coefficients. However, none of these variable selection methods are robust to outliers. Robust versions of the LARS, LASSO and SCAD methods have been considered in the literature (Owen 2006; Khan et al. 2007; Wang et al. 2007; Wang and Li, 2009; Arslan 2012 and Alfons et al. 2013).

In general all these previous methods are prone to getting trapped in locally optimal models and face design problems with complex patterns of multicollinearity, specifically in large datasets (Hans and Dobra 2007). In order to avoid these shortcomings several metaheuristic techniques have been developed for solving large problems, such as Simulated Annealing (Meiri and Zahavi 2006) and Genetic algorithm (Kilinc et al. 2016; Zhu et al. 2017).

More recent works have applied these methods to real data; for example Hasan et al. (2015) and Sun et al. (2016). Some works also cover other prediction and/or classification models, for example Kim and Hong (2017), Bouveyron and Jacques (2010), Genuer, et al (2010) and Ma et al (2006).

The majority of methods on variable selection for regression are heuristic techniques, such as those mentioned above. Nevertheless, exact methods have also been developed to obtain optimum solutions in small or medium-size problems; for example, Gatu and Kontoghiorghes (2006), Gatu et al (2007) and Brusco et al (2009). An example of exact variable selection methods in other prediction and/or classification models may be found in Brusco and Steinley (2011).

As has been seen, there are very many references of works in which the problem of variable selection in linear regression models are analyzed. Nevertheless, as commented in sub-section 1.1, in none of these works do we find the specific problem of variable selection that is analyzed in this work (and that has been explained and reasoned in sub-section 1.1). To our knowledge, this variable selection problem has only previously been analyzed in Pacheco et al (2013), albeit in the context of PCA. No references on either regression or classification models have been found.

1.3. *Contribution*

In this work, an exact method for the variable selection problem in regression models (described in sub-section 1.1), is proposed. Also, different tools and strategies are proposed for improving its computation time such as: use of filters or pre-selection to avoid unnecessary explorations, and the use of previous information obtained by means of the execution of a simple and fast heuristic.

A set of computational experiments have been executed, which demonstrates the high efficiency of these tools. Indeed, the computational time is reduced significantly with these tools and, therefore, the size of problems that can be solved in moderate time is increasing.

In summary, the main contributions of the work are as follows: 1) an analysis of a new variable selection problem in regression. This problem has important applications, specifically in the field of composite indicators. 2) The design of an exact method to find optimal solutions to this problem in moderate-size datasets. 3) The incorporation to this exact method of some strategies that reduce computational times. It should be pointed-out that the first of these strategies (filter or preselection) is an "ad-hoc" design for this specific problem. Nevertheless, the second one, (the use of previous information obtained by a fast heuristic), could be easily used once adapted to other problems (not only variable selection, but also location problems, for example).

The remainder of this work is organized as follows: in section 2, the definition of the problem is outlined. In section 3, the basic Branch & Bound method is explained and the different tools to accelerate the Branch and Bound method are analyzed in Section 4. Section 5 contains a description of the simple and fast heuristic method. The computational experiences are shown in section 6. In section 7, an example based on real data is proposed. The last section presents the final conclusions of the study and our related future research lines.

2. Definition of the Problem

2.1. *Prior Definitions*

Consider a data matrix, *X*, corresponding to *m* cases and characterized by *n* variables. We shall label the set of variables $V = \{1, 2, \dots, n\}$ (the variables are identified by their indices for the sake of simplicity).

Let x_{ij} be the value of variable *j* in the case *i*, *i*=1,…,*m*; *j* = 1,…,*n*; Let x_j be the column vector with the values of variable *j*, in other words

$$\boldsymbol{x_j} = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{mj} \end{pmatrix} \qquad j = 1, \dots, n.$$

It is known that $X = \left(x_{ij}\right)_{i=1,\dots,m;j=1,\dots,n} = (\boldsymbol{x_1} \quad \boldsymbol{x_2} \quad \cdots \quad \boldsymbol{x_n})$

Let y_i be the value of a variable y in the case i, $i = 1, \dots, m$;

For any subset of variables $S \subset V$, let's define

$f(S)$ = R-squared (R^2) value of the linear regression model with y as the dependent variable and S as set of independent variables.

2.2. *Formulation of the Problem*

Consider a partition in q groups of the set of variables V, that is $V = \bigcup_{r=1}^{q} G_r$; where G_r represents each group of variables into which V is divided. Let $p \in N$, verifying $1 \leq p \leq n$, so that the problem may be defined as:

$$\text{Maximize } f(S) \tag{1}$$

subject to:

$$|S| = p \tag{2}$$

$$S \cap G_r \neq \varnothing \qquad r = 1, \ldots, q, \quad \text{if } p \geq q$$

$$|S \cap G_r| \leq 1 \qquad r = 1, \ldots, q, \quad \text{if } p < q \tag{3}$$

$$S \subset V \tag{4}$$

The optimum solution and the value corresponding to the problem defined by (1)-(4) are respectively denoted by S_p^* and $g(p)$, that is $g(p) = f(S_p^*)$.

Apparently, there is no real or practical interest in determining the values of $g(p)$ for $p < q$. Nonetheless, these values help to accelerate the execution of the Branch & Bound Method as will be explained in detail in section 3.

As it has been said in the introduction, one of the main applications of this model is the design and/or the update of the composite indicators. Suppose a composite indicator formed by a big set of variables. If the set of variables that formed the composite indicator is too big it could be convenient (both from the economic point of view, and from the point of view of understanding) to reduce the number of variables that explain the indicator, while the approximation (correlation) to the initially obtained indicator is maximized. In other words, the objective is to select a subset of variables of smaller size, which is able to explain most of the information of the initial composite indicator (that is, the one obtained with all the original variables). On the other side, if the set of original variables is composed by groups that reflect the different aspects of the analysed problem (some examples have been described in the introduction) it should be convenient that the subset contains variables of all of these groups.

So, in the previous formulation V is the set of original variables, G_r represents each group of variables that reflects the different aspects of the analysed problem, y is the variable that contains the values of the composite indicator originally obtained and S is the subset of smaller size that must be obtained.

In section 7, an example with real data is given of this problem. In this case, it is a question of analyzing the evolution of Spanish economy, through the different socio-economic variables divided into 6 different groups.

3. Description of the Basic Branch & Bound Method

The corollary in Appendix 1 allows the design of an exact Branch & Bound (*BnB*) based-method, similar to others found in the literature for several variable selection problems (Brusco and Steinley 2011; Pacheco et al. 2013). When p_0 ($p_0 \geq q$, $p_0 \leq n$) is a fixed value, then this method allows us to find the optimal solutions, S_p^* and $g(p)$, for all values of p ($p \leq p_0$), when the value of n is moderate.

The *BnB* algorithm performs a recursive analysis of the set of solutions. This analysis is performed by means of a search tree. Each node of the tree corresponds to a set of solutions. When a node J is explored, the question of whether the set of associated solutions can improve some of the values $g(p)$ found up until that moment is determined. If it is determined that no associated solution to node J can improve any value of g, the exploration of that node is ended. If otherwise, the associated set is divided into two subsets that

are associated with both nodes K and L that emerge from node J. Subsequently, nodes K and L are explored. The node of origin that corresponds to all the solutions is explored first.

More specifically, the solutions associated with each node J are determined by two subsets A and $B \subset V$, such that $A \cap B = \varnothing$. In this way, the set of solutions for J are all subsets $S \subset V$ that contain A ($A \subset S$) and that do not contain elements of B ($B \cap S = \emptyset$). (In other words, the elements of A would be "fixed, and those of B "forbidden"). Division of node J in nodes K and L entails the determination of an element $a \in V - A - B$. Subsequently, the sets $A' = A \cup \{a\}$, $B' = B$, $A'' = A$ and $B'' = B \cup \{a\}$ are defined and then nodes L and K are respectively associated with the solutions determined by A'' and B'' (L) and A' and B' (K). Figure 1 illustrates the functioning of this recursive division.

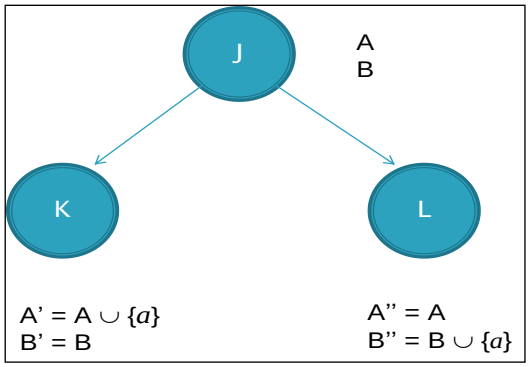

Figure 1. Branch process in the Branch & Bound method

Let A and B be two subsets $A, B \subset V$, such that $A \cap B = \emptyset$; a description in pseudocode of the exploration of each node associated with them is as follows:

Procedure *ExplorationNode* (A, B)

If $\exists\, r \in \{1, \ldots, q\}$: $V - B \cap G_r = \emptyset$, then Exit (finalize Exploration of the node) (6)

If $f(A) > g(|A|)$ and A is a *feasible* set then make $S^*_{|A|} = A$ and $g(|A|) = f(A)$ (7)

If $f(V-B) > g(|V-B|)$ then make $S^*_{|V-B|} = V - B$ and $g(|V-B|) = f(V-B)$

If $(|A| = p_0)$ or $(A \cup B = V)$ then Exit (finalize Exploration of the node)

If $g(|A|) \geq f(V-B)$ then Exit (finalize Exploration of the node) (8)

Determine $a = argmax\, \{ f(A \cup \{v\}) / v \in V - A - B \}$ (9)

Make $A' = A \cup \{a\}$ and $B' = B$

Execute Exploration_node (A', B')

Make $A'' = A$ and $B'' = B \cup \{a\}$

Execute Exploration_node (A'', B'')

Pseudocode 1. Procedure *ExplorationNode*

Thus, the *BnB* method can be described in the following way

Method *BnB*

Make $g(p) = 0, \forall\, p \leq p_0$ (10)

Make $A = \emptyset$, $B = \emptyset$

Execute *ExplorationNode*(A, B)

Pseudocode 2. Method *BnB*

Finally, the following point should be made:

- A set A is *feasible*, in line (6), if the following condition is fulfilled:
  if $|A| \geq q$ then $A \cap G_r \neq \emptyset$ $\quad \forall\, r = 1, \ldots, q$
  if $|A| < q$ then $|A \cap G_r| \geq 1$ $\quad \forall\, r = 1, \ldots, q$
- The restriction that is asked for in line (8) of the *Exploration_node* procedure ensures that there are no solutions in that node that will improve the values of g and the exploration should therefore be ended. This is based on the corollary from sub-section 3.1. In fact, it follows that any solution S in that node verifies $A \subset S \subset V - B$. Therefore, if the restriction in line (8) is fulfilled, then $f(S) \leq f(V - B) \leq g(|A|) \leq g(|S|)$; so, S will not improve $g(|S|)$.
- It has to be said that, although S_p^* and $g(p)$ are respectively defined as the optimum solution and its corresponding value to the problem (1) – (4) (sub-section 2.2), in the description of the algorithm they are the corresponding approaches found during the search. Obviously at the end of the execution of the *BnB* method S_p^* and $g(p)$ correspond with this optimum and its objective function value.
- As may be confirmed in pseudocodes 1 and 2, the algorithm solves the problem (1) – (4) for all the values of p such that $p \leq p_0$. It must be pointed out that all the values $g(p)$, $p \leq p_0$ (including those corresponding to $p < q$) are important for the algorithm to function properly. In fact, high values of $g(p)$ permit the restriction in line (8) of the ExplorationNode procedure to be met, and unnecessary explorations are therefore avoided. If only the value of g(p0) is updated, but the values $g(p)$ for $p < p_0$ are not updated, then $g(p) = 0$ will remain true, for $p < p_0$. So, this restriction may never be satisfied and therefore the exploration of the corresponding node will have to continue, even though it contains no reliable solutions. Therefore, to avoid high computation time, it is important that the algorithm updates all the values of g(p), for $p \leq p_0$ (including those corresponding to $p < q$), even though our final interest is only to determine $S_{p_0}^*$ and $g(p_0)$. In this sense, a strategy to reduce the computation time (that will be explained in more detail in section 4) is not to start the algorithm with $g(p) = 0$ in line (10), but with good approximations to $g(p)$ and S_p^*. Concretely, it will be proposed as initial values of $g(p)$ and S_p^* those obtained by a rapid heuristic method. In this way, the fulfilment of the restriction in line (8) is favored from the start and unnecessary explorations are therefore avoided. In section 6, through the computational experiments with the different variants of the Branch & Bound method described in section 4, the effect of using this strategy is analyzed.

4. Description of Different Tools and Variants

In order to reduce the computational time of the basic Branch and Bound method some modifications are proposed. These modifications consist in adding certain tools (use of filters and previous heuristics information) and they result in different variants. The modifications and the corresponding variants are described below:

- The restriction of having one element of each group in the solutions that are obtained, can be taken into account when considering the element a to enter. Unnecessary explorations may be avoided if preference is given to the variables of those groups that have no element in A. More formally, a subset of variables $Pre\_Sel \subset V - A - B$ may be formed and a can be chosen in this subset. The subset Pre_Sel (pre-selected variables) is defined as follows:

If $A \cap G_r \neq \emptyset, \forall\, r \in \{1,\ldots,q\}$ then make
$Pre\_Sel = V - A - B$;
otherwise, make
$Pre\_Sel = \{\, v \in (V - A - B) \cap G_r : r \in \{1,\ldots,q\}, A \cap G_r = \emptyset \,\}$.

Therefore, the first variant (*PreSel1*) consists of the substitution of line (9) from the *ExplorationNode* procedure by the following two lines: (9a) and (9b)

Determine the sub-set $Pre\_Sel \subset V - A - B$ (as it is defined above) (9a)
Determine a = argmax { $f(A \cup \{v\})$ / $v \in Pre\_Sel$ } (9b)

The effect of using this alternative form of selecting a will be analyzed in the computational tests.

- The second variant (*PreSel2*), is very similar to *PreSel*1. The only difference is that in the variant *PreSel2* the subset *Pre_Sel* is defined as follows:

If $A \cap G_r \neq \emptyset, \forall\, r \in \{1,\ldots,q\}$ then make
$Pre\_Sel = V - A - B$;
otherwise, make
$Pre\_Sel = \{\, v \in (V - A - B) \cap G_{rmin} \,\}$
where $rmin = argmin\,\{\, |(V - A - B) \cap G_r| : r \in \{1,\ldots,q\}, A \cap G_r = \emptyset \,\}$

In these two variants, preference is given to the groups without elements in A, in case such groups are found. In the variant $PreSel1$, the elements of $V - A - B$ of all these groups are included in Pre_Sel. In variant *PreSel2* the elements of the group with the fewest elements in $V - A - B$ among those groups are included. The idea of this second alternative is "to force fill", in the nodes of the "right branch", the set B of elements of this group (in other words to remove elements of that group from $V - B$). If this condition arises, the exploration at that node is ended, in line (6), and the number of explorations may be reduced.

As explained earlier, the expression of line (8) can help to identify and to avoid unnecessary explorations. However, the values of $g(p)$ are initially given a value of 0, as indicated in the expression of line (10). This value means that there is no compliance with the condition of line (8) in the first iterations, and the corresponding explorations are therefore not interrupted. Subsequently, compliance with this condition is forthcoming more and more as the values of $g(p)$ are updated and increased.

Therefore, one idea that may help to increase the proportion of times that compliance with the condition of line (8) is forthcoming, and thereby to reduce the explorations, is to find initial values of $g(p)$ as quickly as possible that are as high as possible. In this sense, different heuristic algorithms have demonstrated that they can find good solutions to variable selection problems. Among the most recent references, the works of Pacheco et al (2009), Brusco et al (2009) and Brusco (2014) may be mentioned. In addition, a much shorter computing time is required by these heuristics methods than the time required by the exact methods. So that, the heuristic strategies can be good options to obtain good initial (high) values of $g(p)$ (and the corresponding approximations to S_p^*).

Moreover, the execution of a heuristic method can give further useful information to be used in an efficient way in the execution of the Branch & Bound method. Specifically, this information may be used to select element a in line (10) for ramification. In particular, the proposition is to determine $\forall a \in V$.

$$maxv(a) = max\{f(S): a \in S, |S| = p_0,\ \ S \text{ solution visited in the execution of the heuristic}\}$$

These values were found during the execution of the heuristic. Subsequently, in the execution of the Branch & Bound method, the element $a \in Pre\_Sel$ with a higher $maxv(a)$ in each exploration is chosen in line (10). In doing so, the calculation of the f function to determine this element is not necessary, at the

same time as a logical rather than an arbitrary criterion is employed. In fact, the elements that belong to the solution $S^*_{p_0}$ obtained by the heuristic will be selected in the first explorations.

Thus two new variants, (named *InfHeur1* and *InfHeur2*) are proposed. These variants simultaneously combine the use of the Pre_Sel sub-set and the use of information provided by a heuristic, as has been explained. So variant *InfHeur1* consists of the two following modifications:

- In line (10) of the *BnB* method, substitute:

  Make $g(p) = 0$, for $p \leq p_0$

  by

  Read the values of $g(p)$ and the corresponding S^*_p values obtained by the heuristic method

- Substitute line (9) of the *ExplorationNode* procedure by the following two lines: (9a) and (9b)

  Determine the sub-set $Pre\_Sel \subset V - A - B$ as defined in variant *PreSel1* (9a)
  Determine $a = argmax\ \{\ maxv(v)\ /\ v \in Pre\_Sel\ \}$ (9b)

The variant *InfHeur2* is very similar to variant *InfHeur1*. The variant *InfHeur2* consists of the same two previous modifications, but it determines the sub-set Pre_Sel (in line 9a) as defined in variant *PreSel2*; that is

Determine the sub-set $Pre\_Sel \subset V - A - B$ as defined in variant *PreSel2*. (9a)

In summary, the difference between *InfHeur1* and *InfHeur2* is the way as this Pre_Sel sub-set is defined and built in step (9a). In *InfHeur1* the Pre_Sel sub-set is defined as in the variant *PreSel1*¸ and in *InfHeur2* the Pre_Sel sub-set is defined as in the variant *PreSel2*. Obviously, the heuristic method should have previously been run to execute these two variants. As it has been mentioned in the introduction, in the following section a heuristic algorithm is described, which will be used to generate this previous information.

In the computational tests, the effect of these modifications (defining and using the Pre_Sel set and using information contributed by a heuristic method) will be examined in section 6.

5. A Simple and Fast Heuristic Algorithm

As commented in section 4, a heuristic method should have previously been run to execute variants *InfHeur1* and *InfHeur2*. We have designed a fast heuristic method to solve the problem defined by (1) – (4), for different values of p ($p \leq p_0$). The Heuristic algorithm, that we propose in this section, obtains the approximations to the values of $g(p)$ (and those corresponding to S^*_p) in a gradual manner; that is beginning with $p = 1$ and ending with $p = p_0$. Also, the solution obtained for $p - 1$ is used as previous information to find the initial solution for p. The set of solutions for the different values of p are kept in the vector $\boldsymbol{S}$, $\boldsymbol{S} = (S^*_1, S^*_2, \dots, S^*_{p_0})$ and the corresponding values of g are kept in $\boldsymbol{G}$, $\boldsymbol{G} = (g(1), g(2), \dots, g(p_0))$. The Heuristic algorithm is described in pseudocode 3.

Heuristic **Algorithm** (input: p_0; var: $\boldsymbol{S}, \boldsymbol{G}$)

1. Determine $i^* = argmax\ \{\ f(\{i\}) : i \in V\ \}$
2. Do $S^*_1 = \{i^*\}$, $g(1) = f(S^*_1)$
3. For p = 2 to p_0 do
   begin
   4. Do $S_{ant} = S^*_{p-1}$
   5. Determine $i^* = argmax\ \{\ f(S_{ant} \cup \{i\}) : i \in V - S_{ant}, S_{ant} \cup \{i\}\ is\ a\ feasible\ set\ \}$
   6. Make $S = S_{ant} \cup \{i^*\}$

7. Execute *LocalSearch*(p, S)
8. Do $S_p^* = S$ and $g(p) = f(S_p^*)$

end

Pseudocode 3. *Heuristic* Algorithm

As may be seen, the heuristic algorithm obtains the initial solution for p = 1, which is trivial. Subsequently, it uses the solution obtained for $p - 1$ (S_{ant}) in each iteration to complete a rapid initial solution S for p. This initial solution is improved by a local search procedure (***LocalSearch***) and by doing so, the approximation to S_p^* and $g(p)$ is obtained.

The procedure ***LocalSearch*** is an iterative method. It works as follows: In each iteration the set of the 'neighborhood solutions' of the current solution S, is explored; if the current solution S is improved by its best neighborhood solution, S', then the current solution moves to S'. The process ends if none of the neighborhood solutions improve the current solution. The set of the 'neighborhood solutions' of the current solution S is denoted $N(S)$. The procedure ***LocalSearch*** is described in Pseudocode 4.

Procedure *LocalSearch*(input: p_0, var S)
Repeat
1. $f_{old} = f(S)$
2. Determine $S' = argmax\{ f(S'') : S'' \in N(S) \}$
3. If $f(S') > f(S)$ then do $S = S'$

until $f(S') \leq f_{old}$

Pseudocode 4. Procedure *LocalSearch*

The set $N(S)$ is the set of *feasible* solutions which can be reached from S by neighborhood moves (in this way, the neighboring moves are identified with the solutions that they generate). In this case, each move is defined by the exchange of an element of S by an element outside it. The concept of the *feasible* solution is established in section 3.

6. Computational Experiences

In order to analyze the performance of the basic Branch and Bound method and its variants a set of computational experiments have been executed. So that, we can also analyze the efficiency of the proposed tools to reduce the computational time of the Branch and Bound method (definition and use of the Pre_Sel subset and use of information provided for a heuristic).

In order to perform these computational experiments a set of the matrices X and y have been designed. The process of the design these matrices are described in subsection 6.1. In sub-section 6.2 the computational experiences and the corresponding results are described.

It should be indicated that all the algorithms, methods and procedures that have been described in this work were implemented in Object Pascal using the Delphi compiler and the development environment Rad Studio (XE10 – Seattle). All the experiments were performed on an i7 4790 CPU 3.6 GHz PC using the same compiler.

6.1. *Design of Data Matrices*

A series of data matrices have been generated for the different computational tests. These matrices are composed of the X matrix of the independent variables, and for the dependent variable (column) y. The process of generating these matrices (similar to those used in Brusco et al 2009, and Pacheco et al 2013) consists of designing population correlation matrices L with size n; a set of m vectors following the normal

distribution with the L correlation matrix is generated from each population correlation matrix L, these m vectors compose the X matrix (every vector is a row) and finally the y column is obtained from X.

The method of generating vectors of a certain multivariate normal distribution of order n, 0 means, and the correlation matrix L, that is $N(\mathbf{0}, L)$, is as follows. A lower triangular matrix T, of order n, is calculated such that $T \cdot T' = L$, subsequently row-vectors $\boldsymbol{z}$ with distribution $N(\mathbf{0}, I_n)$ are generated, and $\boldsymbol{x} = \boldsymbol{z} \cdot T$ is calculated; the $\boldsymbol{x}$ vectors calculated in this way follow the distribution $N(\mathbf{0}, L)$.

There are several ways of obtaining the lower triangular matrix T such that $T \cdot T' = L$. In our case, the square root method was used, which we find in works by Naylor (1977) and Rubinstein (1981). Also, different methods may be found in these texts to generate the values of a normal distribution $N(0,1)$.

The population correlation matrices L are designed according to a simple pattern: the correlations between the different variables can have two values: a high value, if they belong to the same group, or otherwise a low one. The population correlation matrices therefore depend on n and q and on the following parameters:

- $sizegroup$: size of each group (to simplify, let us suppose groups of the same size, and then $sizegroup = n/q$)
- $wcor$: correlation between variables of the same group (let us suppose that this correlation is higher than between the variables of different groups)
- $lcor$: correlation between variables of different groups.

The following values will be set for the tests: $lcor = 0.2$; $wcor = 0.7$. Besides, $sizegroup$ will take two values: $sizegroup = 3$ and $sizegroup = 5$.

As has been explained above, for every correlation population matrix L, a set of m vectors, (cases), are generated following the distribution $N(\mathbf{0}, L)$. A value of m = 100 was used. These m vector (cases) compose the matrix X.

Finally, the values of y_i , $i = 1, \dots, m$, are obtained in the following way:

$$y_i = \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i2} + \cdots + \beta_n \cdot x_{in} + 0.5 \cdot \varepsilon$$

where, ε is a vector generated from the normal distribution $N(0,1)$.

The values of β_i are distributed in the same way in all the groups. Specifically, in the groups of $sizegroup = 3$ the values are distributed as follows:

$$\beta_1 = 0, \beta_2 = 0.3, \beta_3 = 1, \beta_4 = 0, \beta_5 = 0.3, \beta_6 = 1, \dots$$

and so on.

In the same way, in the groups of $sizegroup = 5$, the values are distributed as follows:

$$\beta_1 = 0, \beta_2 = 0, \beta_3 = 0.3, \beta_4 = 0.3, \beta_5 = 1, \beta_6 = 0, \beta_7 = 0, \beta_8 = 0.3, \beta_9 = 0.3, \beta_{10} = 1, \dots$$

and so on.

Observe that for each matrix X, the partition of the set of variables V is also established, in disjoint sets, ($V = \cup_{r=1}^{q} G_r$), according to the values of q and $sizegroup$, as previously explained. So, if $sizegroup = 3$ then $G_1 = \{1,2,3\}$, $G_2 = \{4,5,6\}$,, and so on; if $sizegroup = 5$, then $G_1 = \{1,2,3,4,5\}$, $G_2 = \{6,7,8,9,10\}$, and so on.

Finally, 7 types of matrices, with different values of q and $sizegroup$, are considered. These values are shown in table 1.

Table 1. Groups of matrices

Type #	q	$sizegroup$
1	7	3
2	8	3
3	9	3
4	5	5
5	6	5
6	7	5
7	8	5

A total of 10 matrices $(X \mid y)$ were generated randomly for each type. These matrices are used in the computational experiments described in sub-section 6.2.

The parameters related with the design of the population matrix, X, follows the same structure and values of Pacheco et al, (2013). The parameters of the linear model used to obtain the vector y follows similar patterns of other recent works, (for example Gijbels and Vrinssen, 2015).

6.2. *Analysis of Basic Branch & Bound Methods and Variants*

In this section the performance of the Branch & Bound method and its variants are compared by using a set of computational experiments. Specifically, all these methods have been executed for each and every matrix that is generated, as described in sub-section 6.1. The way in which the corresponding set of variables V is divided in disjoint sub-sets G_r has also been described in sub-section 6.1. In all cases, the value $p_0 = 2 \cdot q$ is used.

The methods that are analyzed and compared in this section are as follows: the basic Branch & Bound method (*BnB*) and its variants *PreSel*1, *PreSel*2, *InfHeur1* and *InfHeur2*. It should be remembered that as these are exact methods, the solutions that are obtained (the values of $g(p)$ and the corresponding S_p^* values) are always optimal. Therefore, the differences in the efficiency of the different methods should be measured by the calculation time that is employed. In table 2, the time employed in seconds for each method is shown for each matrix (*Time*). The symbol "*" indicates, for each matrix, the exact variant with the shortest calculation time. The following should be pointed out: as stated earlier, before the execution of the variants *InfHeur1* and *InfHeur2*, the *Heuristic* method is executed (described in section 5). A column is therefore added with the execution time of the *Heuristic* method.

Table 2. Computation time of each method

Matrices		***BnB***	***PreSel1***	***PreSel2***	***InfHeur1***	***InfHeur2***	***Heuristic***
Type	**#**						
1	1	8.178	4.552	2.228	2.412	1.585*	0.335
	2	8.185	4.632	3.822	2.803	1.624*	0.336
	3	8.115	4.665	3.315	3.400	2.292*	0.332
	4	6.259	3.566	2.267	2.704	1.548*	0.391
	5	6.266	3.642	2.230^{+}	3.049	1.909*	0.375
	6	7.861	4.227	2.369	2.692	1.597*	0.334
	7	7.004	4.187	4.127	2.485	1.328*	0.335
	8	7.874	4.312	2.053^{+}	3.012	1.880*	0.327
	9	11.824	6.515	3.885	3.352	2.275*	0.383
	10	7.693	4.423	4.289	3.355	2.166*	0.333
2	1	22.807	11.628	14.223	9.786	4.866*	0.586
	2	50.096	28.180	12.324	9.818	6.698*	0.594
	3	29.826	15.149	11.168	6.802	4.292*	0.605
	4	23.448	12.519	10.925	8.907	4.121*	0.572
	5	24.188	12.565	9.902	8.202	4.076*	0.743
	6	29.472	14.907	8.991	9.761	4.925*	0.661
	7	22.262	11.472	5.941	9.349	4.453*	0.588
	8	13.759	8.615	11.675	7.950	3.611*	0.562
	9	44.475	23.268	8.915	9.774	5.096*	0.626
	10	27.871	15.299	11.728	11.045	6.222*	0.788

Matrices		*BnB*	*PreSel1*	*PreSel2*	*InfHeur1*	*InfHeur2*	*Heuristic*
Type	**#**						
3	1	85.655	40.673	29.571	32.526	14.680*	0.985
	2	106.821	49.732	36.333	32.660	15.355*	0.988
	3	146.531	67.591	26.667	31.346	12.437*	1.042
	4	124.232	51.665	60.583	33.418	16.578*	1.074
	5	78.628	40.762	21.754	29.759	9.961*	0.934
	6	80.212	37.923	36.913	31.689	13.519*	0.888
	7	102.018	47.532	30.105	33.532	15.913*	1.042
	8	88.715	41.550	20.376	30.852	13.667*	0.935
	9	109.953	55.109	37.578	36.635	17.559*	1.143
	10	91.688	43.984	41.395	31.560	14.845*	1.088
4	1	23.161	16.566	10.471	8.592	8.273*	0.236
	2	15.441	11.275	11.505	6.850	6.458*	0.275
	3	30.630	24.142	25.916	13.068	12.918*	0.315
	4	19.968	13.595	13.514	7.962	7.678*	0.424
	5	18.795	15.058	15.601	8.335	7.918*	0.398
	6	29.065	22.387	25.963	14.084	14.018*	0.322
	7	12.839	9.000	9.628	4.848	4.505*	0.233
	8	17.448	12.816	12.275	7.605	7.279*	0.352
	9	32.488	25.734	12.883	8.840	8.495*	0.290
	10	28.404	20.857	20.208	7.113	6.928*	0.382
5	1	281.129	229.820	272.928	78.968	77.197*	0.893
	2	115.541	72.087	91.695	26.640	23.972*	0.467
	3	101.412	76.978	90.245	49.066	46.928*	0.442
	4	133.038	93.123	96.948	56.648	55.607*	0.629
	5	109.366	67.413	103.564	41.530	39.736*	0.598
	6	178.352	131.850	78.520	57.972*	59.091	0.705
	7	118.442	68.705	180.158	46.558	45.546*	0.732
	8	165.463	112.672	61.556	52.942	52.750*	0.584
	9	113.971	69.582	96.135	44.940	44.237*	0.456
	10	114.791	74.519	84.633	38.962	35.881*	0.524
6	1	2162.385	1746.558	1059.107	437.480	430.979*	1.021
	2	632.697	403.787	539.758	241.909	232.662*	1.456
	3	977.758	662.249	671.738	663.582	638.168*	1.184
	4	910.269	562.934	727.655	361.445*	365.843	0.988
	5	1367.720	1103.653	1496.795	762.038	732.485*	0.941
	6	737.239	496.135	501.553	396.608	379.949*	1.162
	7	511.822	328.790	266.412	161.346	146.905*	0.877
	8	1061.915	830.588	437.599	332.380	318.450*	0.842
	9	873.878	553.175	212.365	155.497	136.268*	1.325
	10	946.731	652.591	711.091	370.250	358.752*	0.995
7	1	6299.295	4223.729	4626.273	2620.425	2568.623*	1.780
	2	4177.868	2835.552	3212.948	1065.095	1024.255*	2.192
	3	5030.946	2722.917	5334.591	2419.112	2347.922*	1.437
	4	3828.211	2354.243	1538.848	1552.349	1458.714*	1.619
	5	4665.308	2996.868	3193.335	2579.508	2538.410*	1.964
	6	3847.838	2333.476	2830.681	982.735	902.005*	2.728
	7	5290.648	3831.725	2258.020*	4352.498	4309.844	1.964
	8	5050.089	3285.511	6214.758	1529.485	1432.115*	1.425
	9	3872.271	4555.524	9926.658	2480.885	2328.264*	1.833
	10	9782.755	7033.737	6342.715	2942.014	2768.234*	1.425

In table 2 the following points may be seen:

- In all cases, the computation times of the original Branch & Bound method (*BnB*) are improved by all its variants (*PreSel1*, *PreSel2*, *InfHeur1* and *InfHeur2*). In addition, these improvements are relevant: the reduction in the computation time is at least 20% and they manage to reach 90%. There are only five exceptions: in matrix 7 of type 5, matrix 5 of type 6 and matrices 3, 8 and 9 of type 7 the times of the *BnB* methods are lower than those of the *PreSel2* method. The proposed strategies to improve the computation times of the *BnB* method have therefore functioned satisfactorily.
- With regard to the definition and the use of the Pre_Sel set: the variant *PreSel2* appears to use less computation time than *PreSel1* in the smaller-sized matrices (types 1, 2 and 3), while those of a larger size (types 5, 6 and 7) *PreSel1* appear to use less computation times than *PreSel2*.

- Moreover, when these variants (*PreSel*1 and *PreSel2*) are added, the use of information supplied by the heuristic method reduces the computation times even more. In this way, *InfHeur1* improves the times of *PreSel*1 (in all cases except in matrix 3 of type 6 and matrix 7 of type 7) and *InfHeur2* improves the times of *PreSel2* in all cases (except for matrix 7 of type 7).
- Finally, *InfHeur2* achieves the best times between the exact methods (*BnB* and variants) in all case (except in matrix 7 of type 7, in which the best time is obtained by *PreSel2* and the matrix 6 of type 5, in which the best time is obtained by *InfHeur1*). The variants *InfHeur1* and *InfHeur2* require prior execution of the *Heuristic* method. Nevertheless, as observed in the table, the time employed by this method is really irrelevant in comparison with those employed by the different exact methods, (except perhaps in the type 1 matrices). Only in instances 5 and 8 of matrix type 1 is the sum of the time *InfHeur2* and *Heuristic* slightly lengthier than the time employed by *PreSel2* (2.284 and 2.207 of *InfHeur2* and *Heuristic* as against 2.230 and 2.053 of *PreSel2*).

Table 3 and figure 2 show the average results by matrix type.

Table 3. Computation time of each method: mean by matrices type

Matrices Type	*BnB*	*PreSel1*	*PreSel2*	*InfHeur1*	*InfHeur2*	*Heuristic*
1	7.926	4.472	3.058	2.926	1.820	0.348
2	28.820	15.360	10.579	9.139	4.836	0.633
3	101.445	47.652	34.127	32.398	14.452	1.012
4	22.824	17.143	15.796	8.730	8.447	0.323
5	143.151	99.675	115.638	49.423	48.095	0.603
6	1018.241	734.046	662.408	388.254	374.046	1.079
7	5184.523	3617.328	4547.883	2252.411	2167.839	1.837

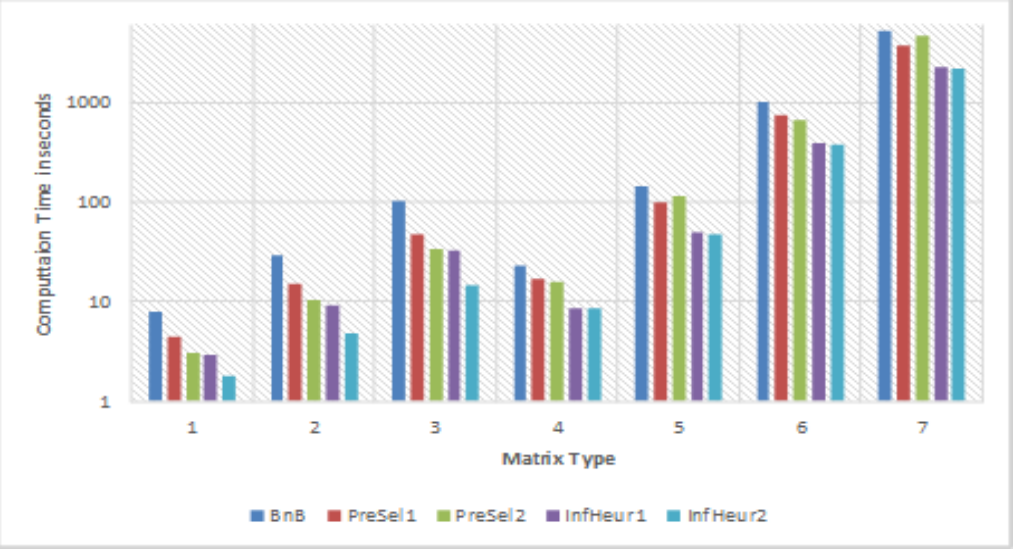

Figure 2. Computation time of exact methods: mean by matrices type

The same conclusions may be reached from table 3 and figure 2 as from table 2: all the proposed variants improve the computation times of the original Branch & Bound (*BnB*) method. Moreover, the variants *InfHeur1* and *InfHeur2* improve the variants *PreSel1* and *PreSel2*. Specifically, the variant *InfHeur2* achieves the best computation times of all the exact methods (even though it includes the additional times of the *Heuristic* method). In short, simultaneously combining both proposed strategies (the use of the *Pre_Sel* sub-set and the use of the information provided by a heuristic) achieve significant and relevant reductions in computation time of the exact original method (*BnB* method).

In table 4 the percentile reductions are shown in the computation time of the four proposed variants with regard to the *BnB* method.

Table 4. Percentile reductions in computation time of the variants with regard to the *BnB* method

Matrix Type	*PreSel1*	*PreSel2*	*InfHeur1*	*InfHeu2*
1	43.426	60.967	62.256	76.679
2	46.470	58.389	64.830	81.974
3	52.761	65.901	67.042	85.397
4	25.522	29.953	60.960	62.412
5	32.232	17.559	64.273	65.385
6	30.316	34.693	60.472	62.057
7	29.755	7.448	55.675	57.384

As may be appreciated, the 4 variants achieve important reductions with respect to the original Branch & Bound method. Nevertheless, in the case of the *PreSel1* and *PreSel2* methods, the reductions appear to decrease as the size of the problem increases. In the case of *PreSel1*, the average reductions vary by 30% (matrix type 6 and 7) to 52% (matrix type 3). Besides, the variant *PreSel2* drops from 60-65% (matrix type 1 and 3) to little more than 7% (matrix type 7). In the case of *InfHeur1* and *InfHeur2*, it is also seen that the reductions lessen as the size of the each matrix increases. Nevertheless, these are maintained even in the larger-sized matrices (matrices type 5, 6 and 7), at around 60% (55-65%).

To end this section, the value of the solutions obtained by the exact methods will be compared with the value of the solution obtained by random selection of the variables. It should be recalled that all the exact methods (*BnB* and variants) obtain the optimum solution for each value of p. Therefore, all the exact methods give rise to the same solution (or solutions with the same value). In table 5, the results are shown of the values of the solutions obtained by the *BnB* method and the values of the solutions that were randomly obtained for $p = p_0$, (where $p_0 = 2 \cdot q$ as defined at the start of this sub-section): specifically the average results by matrix type.

Tabla 5. Comparison of the optimum values (BnB) and those obtained by random selection

Matrices Type	*BnB*	*Random Selection*
1	0.99030	0.93163
2	0.99373	0.95672
3	0.99521	0.94828
4	0.98380	0.89297
5	0.98765	0.90904
6	0.99116	0.92774
7	0.99237	0.92709

As expected, the random selection clearly gives rise to worse results than the optimum solution. In addition, it should be indicated that the randomly generated solution never coincides with the optimum solution. It should be pointed out that the random method has been designed to respect restriction 3 of the problem. Specifically, a variable is chosen at random from each one of the q groups, subsequently $p - q$ variables are chosen, also at random, from among the remaining $n - q$ variables.

7. Analysis of a Real Case

In order to show real applications of the model, an example with real data is commented on below, in an analysis of the evolution of Spanish economy. Specifically, in this case, *Industrial Production Index* (IPI) is taken as an endogenous variable and a set of 38 variables, describing different aspects of the Spanish economic, as the independent variables. These independent variables are divided in 6 different groups. The values correspond to monthly observations in the period 2005-2016 (both included). The table 6 shows the

groups and the variables composing each group. It must be highlighted that these variables and this division have been used in Bujosa et al (2013). Also the real data has been supplied by the Spanish Ministery of Economy and Competitiveness through its web of services (http://serviciosede.mineco.gob.es/Indeco/), specifically in the series of economic conjuncture data base. As it can be observed this website follows the same grouping. Also, as it is explained in Bujosa et al (2013), The Conference Board (TCB) has been working in the design of composite index by using a short list of variables from similar groups.

In all, 144 cases were considered without the seasonal component. With the data described above, it is a matter of selecting the subsets of explanatory variables, verifying that there is at least one variable from each group, which obtains the best fit for R^2. In our case, solutions will be obtained for all values of $p \geq q$. To do so, the *BnB* method was run and its variants for $p_0 = n$. The cases corresponding to 8 first years (2005-2012) have been used to obtain these solutions (i.e. m = 96). The remaining cases are used to observe the *out-of-sample* forecasting performance of the models previously obtained.

Table 6. Variables used in the description of Spanish economy

Group A: Gross fixed capital formation: consumption	1. *Cement consumption* 2. *Construction production index* 3. *Housing starts* 4. *Building permits* 5. *Total houses* 6. *Official licenses* 7. *Official licenses of buildings* 8. *Official licenses of civil buildings*
Group B: Gross value added by industry	9. *Electricity consumption* 10. *Industrial new orders: general* 11. *Industrial new orders: Consumption goods* 12. *Industrial new orders: Intermediate Goods* 13. *Stocks of industrial orders* 14. *Availability of equipment goods*
Group C: Gross value added by services	15. *Air traffic* 16. *Passengers entrance: tourists* 17. *Total tourists* 18. *Overnight accommodation* 19. *Transport of passengers by road* 20. *Indicator services activity* 21. *Fuel consumption* 22. *Workers in SS system: services sector*
Group D: Private consumption	23. *Consumption goods availability* 24. *Real wage indicator* 25. *Car registrations* 26. *Motorcycle registrations* 27. *Retailing sales indicator* 28. *Consumer confidence index* 29. *Home situation in the last 12 months* 30. *Home situation over next 12 months* 31. *Country situation in the last 12 months* 32. *Country situation over next 12 months* 33. *Commercial vehicles registration* 34. *Wage income*
Group E: Labour market: affiliations to SS system	35. *Industry* 36. *Construction* 37. *Agriculture and fishing*
Group F: VAT revenues	38. *VAT revenues*

In table 7, the computation times for these methods are shown. As in the tests detailed in sub-section 6.2, the *Heuristic* method was run before the execution of the variants *InfHeur1* and *InfHeur2*. So, a column with the time (processor-time in seconds) of execution of the *Heuristic* method is also added.

Table 7. Computation times employed with the real data

Computational Time	***BnB***	***PreSel1***	***PreSel2***	***InfHeur1***	***InfHeur2***	***Heuristic***
	8996.628	4652.254	4321.578	3562.23	2945.562	7.847

As may be appreciated in table 7, the exact methods (*BnB* method and its variants) expend and important amount of computational time. In any case these computational times corresponds with the computational times show in section 6.2 in similar size matrices. In table 8 and figure 3, the evolution is shown of the optimal values for the different values of $p \geq q$ ($q = 6$).

Table 8. Evolution of the optimal values

p	**Optimal value**	***p***	**Optimal value**	***p***	**Optimal value**
6	0.98751	**17**	0.99494	**28**	0.99631
7	0.98982	**18**	0.99507	**29**	0.99641
8	0.99084	**19**	0.99525	**30**	0.99645
9	0.99175	**20**	0.99546	**31**	0.99647
10	0.99248	**21**	0.99560	**32**	0.99648
11	0.99296	**22**	0.99567	**33**	0.99649
12	0.99339	**23**	0.99579	**34**	0.99649
13	0.99396	**24**	0.99590	**35**	0.99649
14	0.99431	**25**	0.99597	**36**	0.99649
15	0.99456	**26**	0.99611	**37**	0.99649
16	0.99478	**27**	0.99622	**38**	0.99649

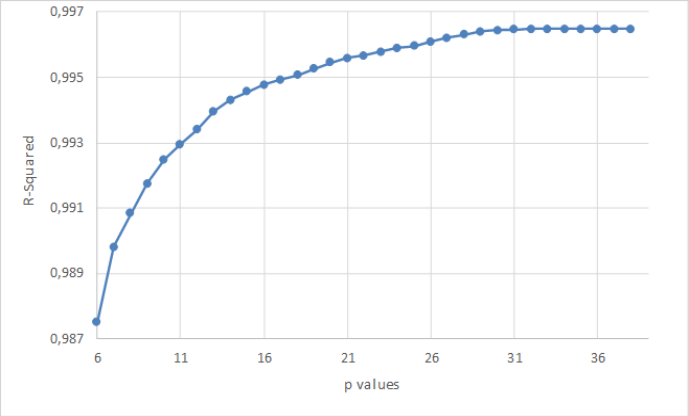

Figure 3. Evolution of the optimal values

Both in table 8 and, above all, in figure 3, a relatively important "leaps" or improvements may be seen from p = 6 to p = 11 in the adjustment (from 0.98751 to 0.99296). From p = 11 the improvements are insignificant. Therefore, if our intention is to look for models that on the one hand have a good adjustment, and on the other are simple, (in other words, have a moderately small number of explanatory variables), the most convenient options appear to be the optimal solutions corresponding to p = 8, 9, 10 and 11. In fact, they imply choosing less than the third part of the original variables and their degree of adjustment is very similar to that obtained with all of the 38 original variables. Table 9 shows the variables that compose these 4 optimal solutions (to simplify we denote them by S^*)-

Tabla 9. Variables in the optimal solutions

p	$R^2 = (f(S^*))$	Variables in S^*
8	0.99084	2 10 14 19 23 24 36 38
9	0.99175	2 10 14 18 19 23 36 37 38
10	0.99248	5 10 13 14 18 19 23 36 37 38
11	0.99296	5 10 13 14 18 19 23 26 36 37 38

As may be confirmed, in fact, at least one variable from each group has been tested, in these p values. It can be observed the following variables appear in the four solutions: *Industrial new orders: general* (10), *Availability of equipment goods* (14), *Transport of passengers by road* (19), *Consumption goods availability* (23), *Construction* (36) and *VAT revenues* (38).

As it has been commented above the cases corresponding to the period 2013-2016 are used to observe the *out-of-sample* forecasting performance of the models previously obtained. Specifically figure 4 compares the evolution of these models with the evolution of the dependent variable IPI.

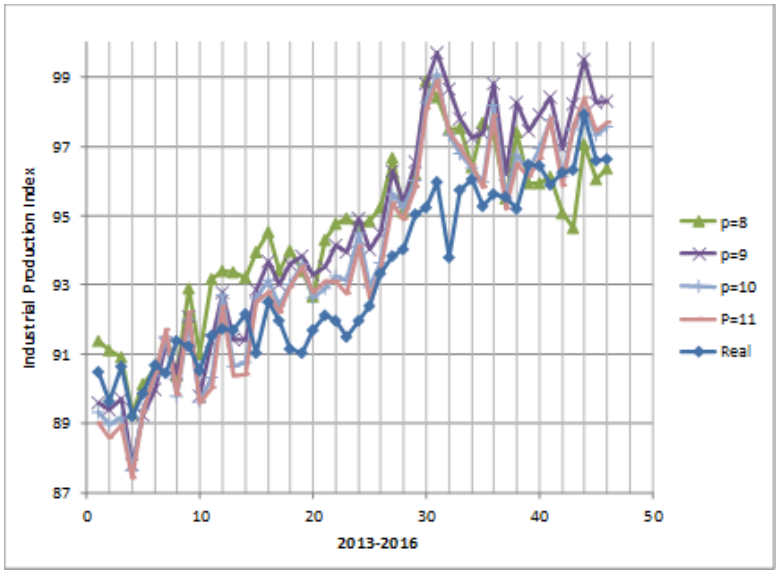

Figure 4. Evolution of the Spanish *Industrial Production Index* and the obtained models in the period 2013-2016

Also an adaptation of LASSO method has been implemented for this model. As it has been indicated in section 1.2 LASSO is based on selection variables by penalizing the coefficients. Figure 5 shows the results obtained by or BnB method, the LASSO adaptation and the Random Selection. The Branch and Bound method obtains the best results for all values of p; except for the largest ones in which ones the three methods obtain the same results.

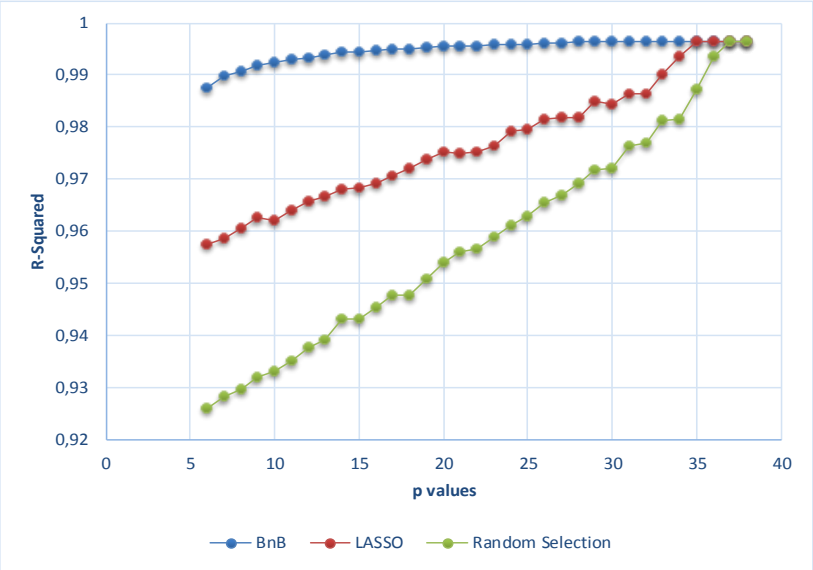

Figure 5. Evolution of the values of solutions obtained by Branch & Bound method, LASSO adaptation and Random Selection

8. Conclusions and Future Research

In this work, a special variable selection problem for linear regression is analyzed. Specifically, the set of original independent variables is partitioned into disjoint groups and the set of variables that is selected should contain elements from all the groups. To the best of the authors' knowledge there are no references in the literature about this specific variable selection problem in the context of linear regression.

This model has a wide field of applications, for example in building composite indicators. The composite indicators should try to cover all points of view of the analyzed issue (economy, society, quality of life, nature, technology, etc.). Each of these different points of view can be identified with a group of variables. So that, the composite indicators should contain at least one variable from each group.

In this work, a Branch & Bound method has been proposed to obtain optimum solutions. As well as having analyzed in detail some strategies and ideas that can accelerate this method (i.e. reduce its calculation time). Specifically, two strategies have been considered: the first one proposes "preselecting those variables, in the branching process, that "help" to fulfill the restrictions of the model; the second one proposes to use the information provided by a previously executed fast heuristic. In the computational tests, it is noted that both strategies combined are efficient. Important and significant reductions in calculation time are achieved, specifically in the largest analyzed problems.

Next, we explain our future related research:

- The heuristic algorithm used in this work is very simple, as may be seen in section 4. In future works, the question of whether more sophisticated heuristics can obtain even more precise information could be analyzed and whether this information can bring about an even larger reduction in the computational time.
- One important shortcoming of our proposed methods is inherent to all exact methods: the analysis of the evolution of computing times indicates that for problems with more than 45-50 variables these times could be excessive. This is because of the variable selection (or feature selection) problems are NP-Hard, as explained in Introduction. In future works, the design of different methods based on metaheuristic strategies that may be used in larger-size problems will be analyzed.

Aknowledgements

This work was partially supported by FEDER founds and the Spanish Ministry of Economy and Competitiveness (Project ECO2013-47129-C4-3-R), the Regional Government of "Castilla y León", Spain (Project BU329U14) and the Regional Government of "Castilla y León" and FEDER founds (Project BU062U16.). These supports are gratefully acknowledged.

Referencias bibliográficas

1. A. Alfons, C. Croux and S. Gelper, Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *Annals of Applied Statistics* **7, 1** (2013), 226-248.
2. O. Arslan, Weighted LAD–LASSO method for robust parameter estimation and variable selection in regression. *Computational Statistics & Data Analysis* **56, 6** (2012), 1952-1965.
3. R. Bandura, A Survey of composite indices measuring country performance: 2008 Update. Office of Development Studies. United Nations Development Programme, *Working Paper* (2008).
4. F.J. Blancas Peral, M. Gonzalez Lozano, F.M. Guerrero Casas and M. Lozano Oyola, Indicadores Sintéticos de Turismo Sostenible: Una aplicación para los destinos turísticos de Andalucia. *Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA, Rect@* **11** (2010), 85-118.
5. C. Bouveyron & J. Jacques, Adaptive linear models for regression: improving prediction when population has changed. *Pattern Recognition Letters*, **31, 14** (2010), 2237-2247.
6. L. Breiman, Better subset regression using the nonnegative garrote. *Technometrics* **37, 4** (1995), 373-384.
7. M.J. Brusco, A comparison of simulated annealing algorithms for variable selection in principal component analysis and discriminant analysis. *Computational Statistics & Data Analysis* **77** (2014), 38-53.
8. M.J. Brusco, R. Singh and D. Steinley, Variable neighborhood search heuristics for selecting a subset of variables in principal component analysis. *Psychometrika* **74** (2009), 705-726.
9. M.J. Brusco and D. Steinley, Exact and approximate algorithms for variable selection in linear discriminant analysis. *Computational Statistics & Data Analysis* **55, 1** (2011), 123-131.
10. M. Bujosa, A. García-Ferrer and A. de Juan, Predicting Recessions with Factor Linear Dynamic Harmonic Regressions. *Journal of Forecasting* **32** (2013), 481–499.
11. Y.K. Chan, C.C.A. Kwan and T.L.D. Shek, Quality of life in Hong Kong: the CUHK Hong Kong quality of life index. *Social Indicators Research*, **71** (2005), 259-289.
12. C. Cotta, C. Sloper and P. Moscato, Evolutionary search of thresholds for robust feature set selection: Application to the analysis of microarray data. *Lecture Notes in Computer Science* **3005** (2004), 21-30.
13. B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, Least angle regression. *Annals of Statistics* **32, 2** (2004), 407-499.
14. J. Fan and R. Li, Variable selection via non concave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96** (2001), 1348-1360.
15. I.E. Frank and J.H. Friedman, A statistical view of some chemometrics regression tools. *Technometrics* 35, 2 (1993), 109-135.
16. W.J. Fu, Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics* **7, 3** (1998), 397-416.
17. C. Gatu and E.J. Kontoghiorghes, Branch-and-bound algorithms for computing the best-subset regression models. *Journal of Computational and Graphical Statistics* **15** (2006), 139-156.
18. C. Gatu, P. Yanev and E.J. Kontoghiorghes, A graph approach to generate all possible regression submodels. *Computational Statistics & Data Analysis* **52, 2** (2007) 799-815.
19. R. Genuer, J.M. Poggi & C. Tuleau-Malot, Variable selection using random forests. *Pattern Recognition Letters*, **31, 14** (2010), 2225-2236.

20. I. Gijbels and I. Vrinssen, Robust nonnegative garrote variable selection in linear regression. *Computational Statistics & Data Analysis* **85** (2015), 1-22.
21. C. Hans, A. Dobra and M. West, Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, **102, 478** (2007), 507-516.
22. M.A. Hasan, M.K. Hasan and M.A. Mottalib, Linear regression-based feature selection for microarray data classification. *International Journal of Data Mining and Bioinformatics* **11, 2** (2015), 167-179.
23. R. Hocking, The analysis and selection of variables in linear regression. *The Annals of Statistics* **32, 1** (1976), 1-49.
24. J.A. Khan, S. Van Aelst. and R.H. Zamar, Robust linear model selection based on least angle regression. *Journal of the American Statistical Association* **102, 480** (2007), 1289-1299.
25. B.K. Kilinc, B. Asikgil, A. Erar and B. Yazici, Variable selection with genetic algorithm and multivariate adaptive regression splines in the presence of multicollinearity. *International Journal of Advanced and Applied Sciences*, **3, 12** (2016), 26-31
26. K. Kim and J.S. Hong, A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis. Pattern Recognition Letters, **98** (2017), 39-45.
27. A.M. López-García and R.B. Castro-Núñez, Valoración de la actividad económica regional de España a través de indicadores sintéticos. *Estudios de Economía Aplicada* **22, 3** (2004), 1-21.
28. S. Luo and S. Ghosal, Forward selection and estimation in high dimensional single index models. *Statistical Methodology*, **33** (2016), 172-179
29. J. H. Ma, Y. Leung & J.C. Luo, A highly robust estimator for regression models. *Pattern recognition letters*, **27, 1** (2006), 29-36.
30. R. Meiri and J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research* **171** (2006), 842-858.
31. R. Mundry and C.L. Nunn, Stepwise model fitting and statistical inference: Turning noise into signal pollution. *The American Naturalist* **173, 1** (2009), 119-123.
32. M. Nardo, M. Saisana, A. Saltelli, S. Tarantola, A. Hoffman and E. Giovannini, Handbook on constructing composite indicators: methodology and user guide. OECD Statistics, *Working Paper* 2005/3 (2005a).
33. M. Nardo, M. Saisana, A. Saltelli and S. Tarantola, Tools for composite indicators building. European Commission. Joint Research Centre. *Working Paper* 21682 (2005b).
34. T. Naylor, Técnicas de simulación en computadoras. Limusa (1977).
35. A.B. Owen, A robust hybrid of Lasso and Ridge regression. Technical Report. Department of Statistics, (Stanford University, 2006).
36. J. Pacheco, S. Casado and S. Porras, Exact methods for variable selection in principal component analysis: Guide functions and pre-Selection. *Computational Statistics & Data Analysis* **57** (2013), 95-111.
37. J. Pacheco, S. Casado and L. Núñez , A Variable Selection Method based on Tabu Search for Logistic Regression Models. *European Journal of Operational Research* **199, 2** (2009), 506–511.
38. S.E. Parada Rico, E. Fiallo Leal and O. Blasco-Blasco, Construcción de indicadores sintéticos basados en juicio experto: aplicación a una medida integral de excelencia académica. *Revista Electrónica de Comunicaciones y Trabajos de ASEPUMA, Rect@*, **16** (2015), 51-67.
39. J. Ramajo-Hernández and M.A. Márquez-Paniagua, Indicadores sintéticos de actividad económica: el caso de Extremadura. Análisis regional: el proyecto Hispalink (Cabrer-Borrás, coord.). Mundi Prensa (2001), 301-312.
40. R. Y. Rubinstein, Simulation and the Monte Carlo method, Wiley (1981).
41. B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo and A. Alonso-Betanzos, Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, **118** (2017), 124-139.
42. S. Sun, Q. Peng and X. Zhang, Global feature selection from microarray data using Lagrange multipliers. *Knowledge-Based Systems*, **110** (2016), 267-274.
43. A. Tangian, Analysis of the third European survey on working conditions with composite indicators. *European Journal of Operational Research* **181, 1** (2007) 468-499.

44. R. Tibshirani, Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B* **58, 1** (1996), 267-278.
45. H. Wang, G. Li and G. Jiang, Robust regression shrinkage and consistent variable selection through the LAD-Lasso. *Journal of Business and Economic Statistics* **25, 3** (2007), 347-355.
46. L. Wang and R. Li, Weighted Wilcoxon-type smoothly clipped absolute deviation method. *Biometrics* **65, 2** (2009), 564-571.
47. Y. Zhu, J. Liang, J. Chen & Z. Ming, An improved NSGA-III algorithm for feature selection used in intrusion detection. *Knowledge-Based Systems*, **116** (2017), 74-85.

Appendix A

Corollary
$\forall\, p, p' \in \{1, \ldots, n\}$, if $p < p'$ then $g(p) \leq g(p')$.

Proof:
By simplifying, we can define $p' = p + 1$,
Three cases may be distinguished: 1) $p < q - 1$; 2) $p = q - 1$ and 3) $p \geq q$.
In both cases by simplifying, we can define $S_p^* = \{1, .., p\}$, and $S' = \{1, \ldots, p, p + 1\}$.

- First case: $p < q - 1$

In this case, with no loss of generality, we assume that 1 is an element of group G_1, 2 is an element of group G_2, …, p is an element of group G_p and $p + 1$ is an element of group G_{p+1}. In this way, S_p^* verifies the restriction (3) as $|S_p^*| = p < q$ and $|S_p^* \cap G_r| \leq 1$, $r = 1, \ldots, q$; in other words, it contains at most an element from each group.

Obviously $S_p^* \subset S'$. In addition $|S' \cap G_r| = 1$ if $r = 1, \ldots, p + 1$ and $|S' \cap G_r| = 0$ if $r = p + 2, \ldots, q$. Restriction (3) is therefore satisfied, as $|S'| = p + 1 < q$ and $|S' \cap G_r| \leq 1, r = 1, \ldots, q$.

Therefore:
$$g(p) = f(S_p^*) \leq f(S') \leq max\,\{f(S) : S \subset V, |S| = p + 1, |S \cap G_r| \leq 1, r = 1, .., q\} = g(p + 1)$$

- Second case $p = q - 1$

The demonstration is similar to the first case: with no loss of generality, we assume that 1 is an element of group G_1, 2 is an element of group G_2, …, p is an element of group G_p and $p + 1$ is an element of group G_{p+1}. In this way S_p^* satisfies the restriction (3).

Obviously $S_p^* \subset S'$. In addition $|S' \cap G_r| = 1$ if $r = 1, \ldots, q$. Restriction (3) is therefore satisfied, as $|S'| = p + 1 = q$ and $|S' \cap G_r| \neq 0, r = 1, \ldots, q$.

- Third case: $p \geq q$

Obviously, $S_p^* \subset S'$. In addition, $' \cap G_r \neq \emptyset$, because $S_p^* \cap G_r \neq \emptyset, r = 1, \ldots, q$.

Therefore
$$g(p) = f(S_p^*) \leq f(S') \leq max\,\{f(S): S \subset V, |S| = p + 1, S' \cap G_r \neq \emptyset, r = 1, .., q\} = g(p + 1).$$