

# Optimal row and column ordering to improve table interpretation using estimation of distribution algorithms

E. Bengoetxea · P. Larrañaga · C. Bielza ·  
J.A. Fernández del Pozo

Received: 22 July 2009 / Revised: 19 April 2010 / Accepted: 19 August 2010 /  
Published online: 16 September 2010  
© Springer Science+Business Media, LLC 2010

**Abstract** A common information representation task in research as well as educational and statistical practice is to comprehensively and intuitively express data in two-dimensional tables. Examples include tables in scientific papers, as well as reports and the popular press.

Data is often simple enough for users to reorder. In many other cases though, there are complex data patterns that make finding the best re-arrangement of rows and columns for optimum readability a tough problem.

We propose that row and column ordering should be regarded as a combinatorial optimization problem and solved using evolutionary computation techniques. The use of genetic algorithms has already been proposed in the literature. This paper proposes for the first time the use of estimation of distribution algorithms for table ordering. We also propose alternative ways of representing the problem in order to reduce its dimensionality. By learning a selective naive Bayes classifier, we can find out how to jointly combine the parameters of these algorithms to get good table orderings. Experimental examples in this paper are on 2D tables.

---

E. Bengoetxea (✉)

Department of Computer Architecture and Technology, University of the Basque Country, Paseo Manuel Lardizabal 1, 20018 San Sebastian, Spain  
e-mail: [endika@ehu.es](mailto:endika@ehu.es)

P. Larrañaga · C. Bielza · J.A. Fernández del Pozo  
Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain

P. Larrañaga  
e-mail: [pedro.larranaga@fi.upm.es](mailto:pedro.larranaga@fi.upm.es)

C. Bielza  
e-mail: [mcbielza@fi.upm.es](mailto:mcbielza@fi.upm.es)

J.A. Fernández del Pozo  
e-mail: [jafernandez@fi.upm.es](mailto:jafernandez@fi.upm.es)

**Keywords** Bertin matrices · Estimation of distribution algorithms · Ordering of tables · Conciseness · Parametrization · Selective naive Bayes

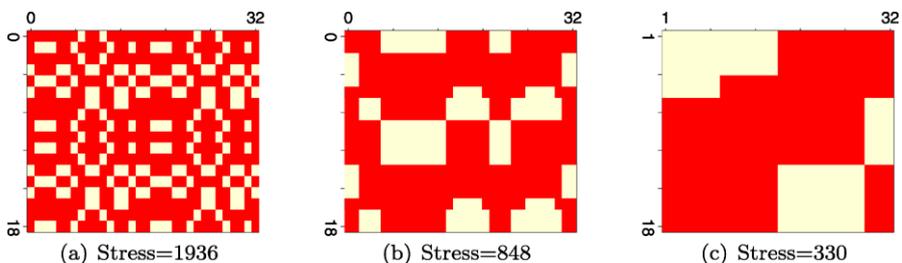
## 1 Introduction

The way rows and columns are ordered in a table is a very sensitive issue that affects the readability of the information. The literature shows that this problem has been a cause of concern in statistics for a long time (Walker and Durost 1936), but is still an important issue nowadays (Koschat 2005). Examples of applications include tables with statistical data for education, practice and research (Koschat 2005), correlation matrices to display microarray data (Friendly 2002), customer satisfaction vs. automobile branches (Niermann 2005) and molecular viewing (Liu et al. 2005).

Rearranging the rows and columns of a table when their ordering is irrelevant reveals interesting patterns that make the table easier to read and interpret. For example, Fig. 1(a) shows a  $36 \times 32$  table, constructed by augmenting 8 times the original  $9 \times 16$  table introduced by Bertin (1981). The columns of this table are townships, while rows are characteristics of these townships, that are either present (coded as 1) or absent (0). The cells with 0s are shaded grey and the cells with 1s are left blank. The row and column order in Figs. 1(a) to (c) varies, showing that the examples with a reduced stress value (Niermann 2005)—see Sect. 2.1—result in a more intuitive readability of the information.

The complexity of the table ordering problem is fully dependent on the size (dimensions) and nature of the information in the table. The decision on how best to order rows and columns is a time-consuming task if tables are big and contain patterns that humans find hard to make out. Indeed, this problem is equivalent to the product of two TSPs (travelling salesman problem). The TSP is known to be an NP-complete problem. As a result of this inherent complexity, metaheuristics should be applied.

Due to the many applications of the table ordering problem, the literature contains many examples of papers applying standard multivariate techniques, such as principal component analysis (Friendly 2002), cluster analysis (Banfield and Raferty 1992) and minimum spanning tree-based algorithms (Friedman and Rafsky 1979), to provide an overall solution for users that need to order rows and columns in the most



**Fig. 1** Different row and column orderings for the same table, illustrating different patterns. Each table ordering is measured according to the stress value defined in Niermann (2005)

comprehensive way. All these methods analyse row permutations and column permutations separately, i.e. considering  $R! + C!$  configurations, where  $R$  is the number of rows and  $C$  is the number of columns. This approach is suboptimal since  $R! \cdot C!$  configurations should be considered. The Linear Ordering Problem (LOP) is a related combinatorial optimization problem with  $R = C$ —where the same permutation of rows and columns is sought such that the sum of the matrix elements above its main diagonal is maximized—which is solved in Garcia et al. (2006) using a variable neighborhood metaheuristic.

Genetic algorithms (GAs) have also been proposed to solve this problem using rather simple crossover and mutation operators (Niermann 2005) or a plethora of operators beyond crossover and mutation (Bielza et al. 2009). In the case of very complex problems, the literature shows that, as the complexity of a problem increases, estimation of distribution algorithms (EDAs), another evolutionary computation paradigm, usually perform better than GAs because they can identify complex patterns within the data (Cesar et al. 2005).

The main contribution of this paper is to propose the use of EDAs and to test their results, firstly through a univariate exploratory data analysis and then using a multivariate approach based on a selective naive Bayes classifier (Langley and Sage 1994). Whereas the first analysis has been routinely applied in the literature (Gómez and Bielza 2004; Larrañaga et al. 1996), the use of a selective naive Bayes approach is another contribution of this paper.

An additional contribution of this paper is to propose new individual representations designed specifically for the table ordering problem. These new representations complement other approaches presented in the literature, such as Niermann (2005) and Bielza et al. (2009), which allow different individuals to have the same fitness value making the optimization algorithms to be confused. This new proposal is designed to better guide the search process when having problems with complex and big tables.

The outline of the paper is as follows. Section 2 illustrates the table ordering problem. It shows how to formalize table ordering as a combinatorial optimization problem. Section 3 is a review of the EDA approach. Section 4 describes the experiments run, the comparative analysis, and the results of applying uni- and multivariate analysis (Sects. 4.3 and 4.4, respectively). Finally, Sect. 5 outlines conclusions and future work.

## 2 Optimal ordering of tables

When displaying a table, rows and columns have to be arranged in the manner that is easiest for the reader to interpret the information provided. Entries that the user wants to compare should ideally be displayed close together, preferably in the same rows or columns. Other arrangements such as sorting rows according to the values of the first column is a good choice for improving table readability (Koschat 2005). However, people's ability to interpret the information in a table is known to be limited by their numerical memory span. This is known to be around seven different digits for most of the population (Miller 1956).

This section formalizes the ordering of rows and columns in a table as a combinatorial optimization problem. This way heuristics such as evolutionary computation can be applied. Two aspects need to be defined: a fitness function of each possible solution and a representation of individuals.

## 2.1 Definition of the fitness function

Since we aim to improve the readability of the tables in order to visually identify definite patterns, the fitness function has to be defined in a way that assigns a better value to more interpretable orderings. Therefore, the clarity of tables has to be evaluated according to conciseness. A way to quantify inconciseness is by computing the distance of each table cell value to its neighboring values, as proposed in Niermann (2005), where a Moore neighborhood is applied. This neighborhood considers the eight neighboring entries of each cell. The resulting *stress* measure is an overall dissimilarity measure for the whole table.

More formally, given a table with  $R$  rows and  $C$  columns, for each table entry  $t_{\pi^r(i), \pi^c(j)}$  and a definite table ordering with row order  $\pi^r = (\pi^r(1), \dots, \pi^r(R))$  and column order  $\pi^c = (\pi^c(1), \dots, \pi^c(C))$ , the Moore neighborhood approach considers that the stress generated by the cell in the  $i$ th row and  $j$ th column,  $s(\pi^r(i), \pi^c(j))$ , is equal to

$$\sum_{l=\max(1, i-1)}^{\min(R, i+1)} \sum_{m=\max(1, j-1)}^{\min(C, j+1)} (t_{\pi^r(i), \pi^c(j)} - t_{\pi^r(l), \pi^c(m)})^2. \quad (1)$$

The total table stress, which is the fitness function, can be computed by adding up the stress of each cell:

$$S = \sum_{i=1}^R \sum_{j=1}^C s(\pi^r(i), \pi^c(j)). \quad (2)$$

Therefore, the table with the minimum stress will be the one with the optimal table ordering, as illustrated in Fig. 1.

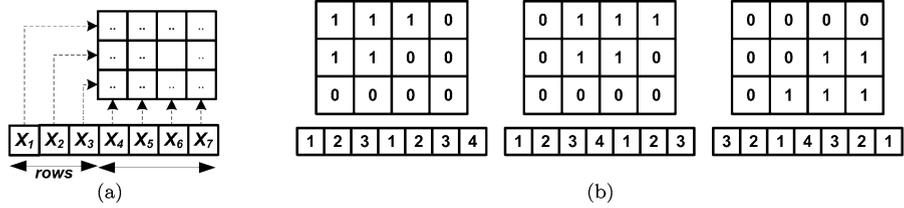
## 2.2 Representation of individuals

This section reviews different possibilities for representing the solutions for the table ordering problem.

### 2.2.1 IRI: Double path representation

Niermann (2005) proposes an individual representation for GAs defined as a definite order number for each of the rows and columns, formed by two concatenated arrays. Similar path-based representations have been applied in the literature to other permutation problems, including the widely known TSP (Larrañaga et al. 1999).

We can formalize this double path-based individual representation as follows. Considering an original reference order for the table with  $R$  rows and  $C$  columns, an individual is defined as  $\mathbf{x} = (x_1, \dots, x_{R+C})$ , where  $x_i = k$  means that the order



**Fig. 2** (a) IR1 individual representation showing the double path representation. (b) Example of different orderings with the same optimum stress value (20) together with the corresponding individuals

of the original  $i$ th row is  $k$  ( $i, k \in \{1, \dots, R\}$ ), and  $x_{R+j} = l$  means that the order for the  $j$ th column is  $l$  ( $j, l \in \{1, \dots, C\}$ ). This was used in Bielza et al. (2009) and is illustrated in Fig. 2(a).

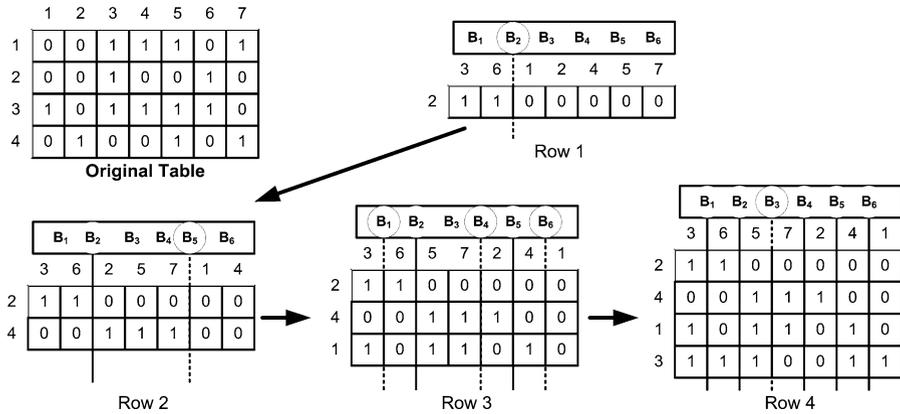
This individual representation has the advantage of being very intuitive for humans, although its simplicity has a drawback: redundancy. Different orderings reflecting table symmetries and rotations will have the same fitness, although they would be represented by very different individuals. For instance, a solution and its transpose would result in the same fitness value. This redundancy confuses the heuristics-based search mechanism (Bengoetxea et al. 2002), since it increases search space dimensionality. Figure 2(b) illustrates this redundancy.

2.2.2 IR2: Single-path representation

We propose a new representation, specifically designed for the table ordering problem. The aim is to get rid of redundant individuals such as the ones generated in IR1.

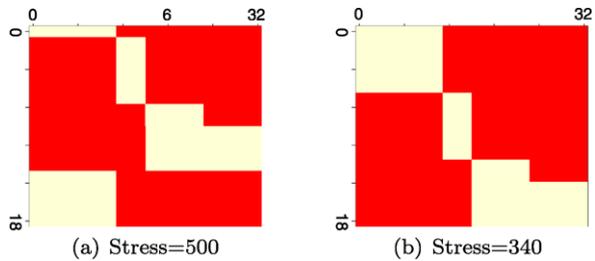
This new individual representation is a single-path representation in which we define only one permutation of the row ordering. The corresponding column ordering is automatically calculated using an encoding which is unique for each individual. The idea is to use a procedure to infer this column ordering from the row ordering and translate from an IR2 to an IR1 representation. The individual will therefore be evaluated using the same fitness function applicable to IR1.

The procedure for inferring the column ordering and transforming an IR2 individual into IR1 form is described next. An example is shown in Fig. 3. The IR1-type solution is built by adding rows one by one in the order described in the individual  $x = (x_1, \dots, x_R)$ , that is, by processing the  $x_l$ th row sequentially for  $l = 1, 2, 3, \dots, R$ . We will also define a Boolean vector  $B = \{B_1, \dots, B_{C-1}\}$ , initializing  $B_b = 0, b = 1, \dots, C - 1$ , to assign borders to column orderings. As shown in Fig. 3, initially we will take the  $x_l$ th row (the second row in the original table) and rearrange columns to place all 1s on the left leaving all 0s on the right. This will activate a border between the columns separating the two types of values ( $B_2$  will be 1) for the consequent rows, although column orders could vary in the following iterations. With the first row processed, and respecting border  $B_2$  for column orderings which divides columns orders in subsets (3, 6) and (1, 2, 4, 5, 7), we will move all 1s as far to the left as possible (see Row 2 in Fig. 3). This could create new column ordering borders ( $B_5$  in the example in Fig. 3). For Row 3, borders  $B_2$  and  $B_5$  should be



**Fig. 3** Example of translating an individual  $x = (2, 4, 1, 3)$  from its IR2 representation into its IR1-type counterpart  $x = (2, 4, 1, 3, 3, 6, 5, 7, 2, 4, 1)$ , given the original table ordering

**Fig. 4** (a) Table obtained using the representation of individuals IR2; (b) using IR3



respected, and therefore only permutations within column subsets (3, 6), (2, 5, 7) and (1, 4) would be allowed—only columns within the same borders—resulting in new borders activated once 1s and 0s have been rearranged. This procedure will continue until all borders have been activated or until all rows have been processed, resulting in an IR1-type solution for which the stress measure defined in Sect. 2.1 can be applied.

Figure 4(a) shows an example of the type of solution that can be obtained using the IR2 individual representation. Note that, whenever possible, individuals always tend to place the 1s in the top-left position.

Figure 2(b) allows us to compare the type of solutions that can be represented by IR1 and IR2, where there is an example of three different orderings that are representing in essence the same solution—they all have the same fitness—even if the row and column orderings taken on a variable by variable basis are quite different. With IR1 all the three can be returned as different individuals, which for EDAs these three individuals are considerably different. In the case of IR2 only the first one could be returned, the other two cannot be obtained with this representation since the automatic column ordering procedure ensures that all 1s will be placed on the top-left corner. In other words, IR2 allows to represent any solution for the table ordering problem (including the optimum one), although it removes the redundancy as to allow a single individual to represent this solution—the only row and column orderings that IR2

does not allow to represent are the ones that do not place the main blocks of 1s in the top-left corner.

The individual representation IR2 ensures that the optimum solution can always be expressed. The only restriction of IR2 over IR1 is that the column order is determined by grouping the 1s in the top-left corner. Since by definition the grouping of 1s is the aim of the optimization problem, all optimum solutions can be represented using IR2.

Regarding the ordering to non-binary tables, the IR2 procedure can also be adapted to these by computing the difference between all values above and below 0.5 and treating them like 1s or 0s in the binary example respectively, which have been normalized previously following the procedure proposed in Niermann (2005).

### 2.2.3 IR3: Local optimization applied to IR2

The use of local optimization techniques in evolutionary computation is very useful for overcoming the particularities of an optimization problem that could not be properly represented through the fitness function.

Figure 4(a) shows that iteratively placing all 1s to the left is not necessarily the ideal arrangement in some cases, and generic local optimizations could easily improve this aspect. For instance, Fig. 4(b) was obtained by applying local optimization to the IR2-type individual in Fig. 4(a). Note that the only difference between (a) and (b) is a small modification in the row ordering. But this results in a significant stress improvement. Local optimization is applied to improve these situations.

Local optimization can be applied to any type of individual representation, without meriting the consideration of different individual representation. In this paper, though, IR3 denotes the application of local optimization by swapping pairs of values to IR2 in order to efficiently describe the experimental results. Every time a new individual is generated, the procedure analyzes all the possible swapping of rows of the IR2-type individual one by one and chooses the best one to replace the original individual.

## 3 Estimation of distribution algorithms

### 3.1 Overview of EDAs

EDAs (Larrañaga and Lozano 2001) are non-deterministic, stochastic heuristic search strategies that form part of the evolutionary computation approaches. In EDAs a number of solutions or individuals are created every generation and evolve repeatedly until a satisfactory solution is achieved. The characteristic that most differentiates EDAs from other evolutionary search strategies such as GAs is that they evolve from one generation to the next by estimating the probability distribution of the fittest individuals and then sampling the induced model. This avoids the use of crossover or mutation operators.

In EDAs the underlying interdependencies among the encoded variables are expressed explicitly through the joint probability distribution associated with the individuals selected at each iteration.

More formally, let  $\mathbf{X} = (X_1, \dots, X_n)$  be a set of random variables, and let  $x_i$  be a value of  $X_i$ . Then, a probabilistic graphical model for  $\mathbf{X}$  is a graphical factorization of the joint generalized probability distribution,  $\rho(\mathbf{X} = \mathbf{x})$  (or simply  $\rho(\mathbf{x})$ ). This can be discrete,  $p$ , or continuous,  $f$ . This model is represented by two components: a structure and a set of local generalized probability distributions.

EDAs have four main steps:

1. Generate the first population  $D_0$  of  $N$  individuals, usually by assuming a uniform distribution (either discrete or continuous) on each variable, and evaluating each of the individuals.
2. Select a number  $Se$  ( $Se \leq N$ ) of individuals (usually the fittest).
3. Induce the  $n$ -dimensional probabilistic model that best expresses the interdependencies between the  $n$  variables.
4. Generate the new population of  $N$  new individuals by simulating the probability distribution learned in the previous step.

Steps 2, 3 and 4 are repeated until a stopping condition is verified. The most important step of this paradigm is to find the interdependencies between the variables (step 3), since methods of learning models from data developed in the domain of probabilistic graphical models need to be adapted to estimate the joint probability distribution associated with the database of the individuals selected from the previous generation.

The model induced in step 3 takes the form of a directed acyclic graph (DAG) that describes a set of conditional interdependencies between the variables on  $\mathbf{X}$ . If  $\mathbf{Pa}_i$  represents the set of parents—variables that are the source of an arrow in the DAG—of variable  $X_i$  in the probabilistic graphical model, the factorization of the joint distribution could be written as

$$\rho(\mathbf{x}) = \rho(x_1, \dots, x_n) = \prod_{i=1}^n \rho(x_i | \mathbf{pa}_i). \quad (3)$$

In the particular case where every variable  $X_i \in \mathbf{X}$  is discrete, the probabilistic graphical model is called *Bayesian network* (Pearl 1988). EDAs in continuous domains assume the joint density function to be a multivariate Gaussian density. The local density function for the  $i$ th variable is computed as the linear-regression model

$$f(x_i | \mathbf{pa}_i) \equiv \mathcal{N}\left(x_i; m_i + \sum_{x_j \in \mathbf{pa}_i} b_{ji}(x_j - m_j), v_i\right), \quad (4)$$

where  $\mathcal{N}(x; \mu, \sigma^2)$  is a univariate normal distribution with mean  $\mu$  and variance  $\sigma^2$ . A probabilistic graphical model built from these local density functions is known as a *Gaussian network* (Shachter and Kenley 1989).

### 3.2 Categorization of EDAs by complexity

EDAs are classified depending on the maximum number of dependencies between variables that they consider (maximum number of parents that a variable  $X_i$  can have in the probabilistic graphical model). Typically, both discrete and continuous EDAs are divided into three categories (Larrañaga and Lozano 2001).

The Univariate Marginal Distribution Algorithm (UMDA) (Mühlenbein 1998) is a representative example of discrete **univariate** EDAs, that is, all variables are considered to be independent. This can be written as  $p_l(\mathbf{x}) = \prod_{i=1}^n p_l(x_i)$ , where  $p_l(x_i)$  is recalculated every generation  $l$  by its maximum likelihood estimation.

In continuous EDAs, the joint density function in this univariate category is assumed to follow a  $n$ -dimensional normal distribution, factorized as a product of  $n$  independent normal densities. An example of continuous EDAs in this category is UMDA<sub>c</sub> (Larrañaga et al. 2000).

In the second category, we have EDAs that can take into account **bivariate** conditional dependencies, that is, each variable can have a maximum of one parent variable. An example of this second category is the greedy algorithm called MIMIC (Mutual Information Maximization for Input Clustering) (de Bonet et al. 1997). The main idea in MIMIC is to describe the true mass joint probability as closely as possible by using only one univariate marginal probability and  $n - 1$  pairwise conditional probability functions. MIMIC<sub>c</sub><sup>G</sup> (Larrañaga et al. 2000) is an adaptation of the MIMIC algorithm to the continuous domain.

The third category is **multivariate** EDAs, where variables can have multiple parent variables, that is, they are able to take into consideration multiple interdependencies between variables. A representative of this category is EBNA (Estimation of Bayesian Network Algorithm) (Etxeberria and Larrañaga 1999). EBNA uses the Bayesian Information Criterion (BIC) (Schwarz 1978) as the score to evaluate the goodness of each structure found during the search. EGNA (Estimation of Gaussian Network Algorithm) (Larrañaga et al. 2000) is an example of a continuous EDA for this third multivariate category. It takes a score + search approach to find Gaussian network structure using a Bayesian score and a local search to find good structures.

### 3.3 Simulation in Bayesian and Gaussian networks

In EDAs, the Bayesian and Gaussian networks are simulated merely as a tool to generate new individuals for the next population based on the previously learned structure. The method usually applied in EDAs is *Probabilistic Logic Sampling* (PLS) proposed in Henrion (1988), where variables are instantiated one at a time in such a way that a variable is sampled after all its parents.

In the case of discrete EDAs, a simple PLS algorithm will not take into account any constraint on individuals set by a particular problem. In the individual representations proposed in the previous section, all representations of a correct individual must contain a permutation of values if they are to represent a solution. That is why we apply the method proposed in Bengoetxea et al. (2002) to obtain only correct individuals that satisfy the particular constraints of our problem.

This method allows using permutation-based representations in EDAs with both discrete and continuous EDAs, although (Bengoetxea et al. 2002) concludes that usually continuous EDAs perform than discrete ones in complex problems. When using continuous EDAs, we propose a strategy based on translating the individual into the continuous domain to a correct permutation in the discrete domain and then applying the same technique as in Bengoetxea et al. (2002). According to this, we propose ordering the continuous values of the individual and setting its corresponding discrete values as the respective order in the continuous individual.

### 3.4 Adapting EDA to solve the table ordering problem

Our proposal is to apply the general EDA approach with no modifications in the learning or simulation phases, and to guide the search for the optimum table ordering using the different individual representations described in Sect. 2.2.

When applying any evolutionary computation technique the main problem of representations such as IR1 is that the redundancy of individuals—having different individuals representing a same solution with the same fitness value—can confuse the search process. This is because the convergence strategy of these algorithms relies on variable values of the most promising individuals, and thus having different and equally promising individuals might confuse the algorithms. This is for instance the reason why evolutionary computation does not perform well in some traditional optimization problems such as 3-SAT ones.

The proposal of IR2 and IR3 as alternatives to the IR1 representation aims at removing this redundancy and at guiding better the search process. However, it must be noted that these two individual representations are based on permutations. Individuals in the form of permutations can be applied in EDAs by adapting their simulation phase if necessary to ensure that all values of the permutation will appear in the final individual (Bengoetxea et al. 2002). As concluded in Bengoetxea et al. (2002).

## 4 Experiments

We describe in this section the experiments performed to test the validity of the EDA approach for improving the readability of two dimensional tables following the fitness function and combining the different choices of individual representations. These experiments also analyse the effect of the different generic EDA parameters on the final result for each of the individual representations. The aim is to suggest a best choice for users not familiar with EDAs.

As previously mentioned, the use of EDAs over other alternatives such as GAs is motivated by the proved ability of EDAs to better identify complex patterns within the data in complex problems (Cesar et al. 2005). The size of the tables chosen for our experiments justifies the use of EDAs due to the added difficulty of identifying patterns. Moreover, these experiments complement the study presented in Bielza et al. (2009) on the analysis of GAs applied to the table ordering problem on the one hand because this time the study is performed for EDAs, and on the other hand because this new study analyses the validity of the new individual representations IR2 and IR3 applied to EDAs.

### 4.1 Description of tables

Of the different examples of tables proposed in the literature for the table ordering problem, we have chosen a typical and simple case given in Bertin (1981). This table, called *Bertin*, was applied in further studies to prove different algorithms and assures results comparability (Niermann 2005). It is illustrated in Fig. 5(a). The table contains arbitrarily ordered characteristics (rows) and townships (columns) with a

(a) Bertin table																
Characteristics	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
High School	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Agricult. Coop.	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
Railway Station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
One-Room-School	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
Veterinary	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0
No Doctor	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1
No Water Supply	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
Police Station	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
Land Reallocation	0	1	1	1	0	0	1	0	0	0	0	1	0	0	1	0

(b) Hospitals table											
ID	BEDS	RBEDS	OUTV	ADM	SIR	H95	K95	TR	H96	K96	FEM
1	220	26	0	5954	9500	793	617	0	768	590	146
2	493	0	44542	12345	8910	80	47	0	65	49	52
3	962	28	721546	37662	23860	69	155	0	107	219	109
4	788	32	209145	22316	7772	108	49	1	97	44	138
5	726	0	58263	26029	20128	132	76	0	150	77	104
6	1048	0	181245	37126	22203	80	0	1	75	0	179
7	565	0	0	18653	13468	62	19	0	67	24	37
8	192	0	0	6333	6314	1421	840	0	1373	727	149
9	634	0	133578	28164	15022	178	68	0	173	77	154
10	788	0	0	26215	2363	126	45	0	133	48	237
11	1071	36	284564	39630	28659	245	68	0	233	70	191
12	1314	41	0	41493	22314	239	155	0	232	119	212
13	212	0	9275	7803	10523	102	284	0	0	0	0
14	829	0	13250	29832	25346	136	62	0	168	53	125
15	103	0	130767	5441	17736	0	0	1	0	0	0
16	543	35	307372	7538	3300	14	13	0	22	20	10
17	1232	52	357425	24875	6295	28	12	1	27	10	85
18	152	152	0	1050	0	0	0	0	0	0	0
19	236	0	0	4938	2998	7	10	0	10	3	33
20	575	20	289584	14645	4338	16	8	0	21	3	18
21	678	30	544529	16071	3722	3	0	1	12	0	52
22	213	0	14855	8526	7215	29	3	0	22	3	56
23	250	60	0	549	0	0	0	0	0	0	0
24	417	120	0	728	0	0	0	0	0	0	0

**Fig. 5** Original tables used in the experimental section. The Bertin table has been augmented several times in our experiments.

binary response (present or absent). With the aim of gaining insights into the scalability of EDAs, the Bertin table was augmented several times in our experiments. These larger tables are denoted as Bertin4, Bertin8, Bertin32 and Bertin128. Additionally, as a non-binary example we chose the Hospitals table proposed in Niermann (2005) to test the table ordering problem. This is a simplification of the original table introduced in Cabrera and McDougall (2002). This table is shown in Fig. 5(b) and contains information on hospitals (rows) versus characteristics (columns) that are discrete (number of beds, visits, operations ...), continuous (administrative cost) and binary (trauma unit). All the different scales measuring these variables are rescaled to the unit interval (see Niermann (2005) for further details). The new rescaled table entries are used to compute the stress of any individual derived during the EDA evolution. Table 1 shows the dimension of the examples we

**Table 1** Dimensions and characteristics of the tables

Table	$R \times C$	Initial $S$	Our best $S$
Bertin4	$18 \times 32$	1936	330
Bertin8	$36 \times 32$	3952	474
Bertin32	$72 \times 64$	16096	978
Bertin128	$144 \times 128$	64960	1986
Hospitals	$24 \times 11$	529	299.27

**Table 2** EDA parameters and their possible values

Parameter	Possible values
<i>EDA type</i>	discrete, continuous
<i>EDA complexity</i>	univariate, bivariate, multivariate
<i>Individual representation</i>	IR1, IR2, IR3
<i>Elitism</i>	yes, no
<i>Population size</i>	$\alpha RC$ , $2\alpha RC$
<i>Selection size</i>	PopulationSize/2, PopulationSize/4

will work with and their corresponding original stress value. Column 4 lists the best results of our experiments.

Note that GAs have been applied in the recent literature for the Bertin and Hospitals examples (Niermann 2005). After being run the same number of times as EDAs on each of the Bertin4, Bertin8, Bertin32, Bertin128 and Hospitals tables, the best stress results for the GA in Niermann (2005) were 346, 550, 2010, 15704, and 342, respectively. Notice that EDAs are able to improve these results for all the example tables.

#### 4.2 Parameterization of EDAs

Table 2 shows a summary of the 6 EDA parameters that are applied in the experiments. This results in a total of  $2 \cdot 3 \cdot 3 \cdot 2 \cdot 2 \cdot 2 = 144$  possible combinations. The *Elitism* parameter refers to how the individuals that are to form the next-generation population are chosen: when activated, the chosen individuals will be the best of the newly simulated and the current population of individuals. In the *Population size* parameter,  $R$  and  $C$  are the number of table rows and columns, and  $\alpha$  serves to adapt the population size to obtain reasonable execution times; in our case,  $\alpha = 1$  for examples Bertin4, Bertin8 and Hospitals;  $\alpha = 0.5$  for Bertin32; and  $\alpha = 0.25$  for Bertin128.

Due to the stochasticity of EDAs, an experiment will consist of 10 executions of each of the 144 combinations of parameters, that is, for each of the 5 examples in Table 1, 1,440 executions altogether.

### 4.3 Univariate analysis of experiments

The summary of the EDA results for the table examples described previously is illustrated in Table 3. Results have been divided into 6 sub-tables, one per parameter. Table 3 includes the mean fitness and standard deviation of the best evaluations found in the executions for each fixed parameter. In each case, the best result for each table ordering example is underlined in bold.

Some interesting conclusions can be drawn from this table. In the case of *EDA type*, it is clear that the best stress is obtained using continuous EDAs. The only exception is the *Hospitals* example, where discrete EDAs performed better. These results confirm the findings comparing discrete and continuous EDAs described in Bengoetxea et al. (2002) when applying individual representations based on permutations to other optimization problems.

Regarding the *EDA complexity* parameter, the best mean results were obtained with bivariate and multivariate algorithms, although multivariate algorithms do not manage to get the optimum value obtained using other EDAs for the *Bertin128* and *Hospitals* examples.

The results for the different individual representation types show that IR2 does not provide good results, although the results are comparable to the classical IR1 representation if local optimization—IR3—is added. A more detailed analysis of the executions shows that IR1 gets the best results more times than IR3 in all the examples, although IR3 arrives at solutions with the better mean stress after the 10 runs. The only exceptions to this are the *Bertin128* example (in which IR3 gets the best results) and the *Hospitals* example (where IR1 appears to behave better, suggesting that IR3 needs some improvement for non-binary tables).

For the *Elitism* parameter, results are very close for the small examples. As the table size increases, however, the non-elitist approach achieves better mean and best results. Results for the *Hospitals* example are similar in both cases, although mean stress results are more positive for the elitist approach (however, this result could vary for bigger non-binary tables).

Finally, the final outcomes of the different combinations of population and selection sizes are quite similar, although the best mean result is obtained in all cases when the selection size is a quarter of the population size.

In the light of these univariate results, the recommendation for the best EDA parameters for *Bertin8* or bigger binary table ordering problems, is to choose a continuous EDA with IR3 or IR1 and use a non-elitist approach and a selection size of a quarter of the population size, preferably with a population size of  $2\alpha RC$ . In the case of non-binary tables, experimental results suggest that the best combination would be to use continuous EDAs and IR1 individual representation, with population and selection sizes of  $2\alpha RC$  and a `population size/4`, respectively.

To get a clearer picture of the effect of some specific EDA parameter choices, we present three sets of histograms. They focus on the choice of the best individual representation and EDA algorithm (according to *EDA type* and *complexity*). Figure 6 shows the performance of the three possible individual representation types for the four binary examples proposed in this paper. Figure 6 clearly confirms that the IR2 individual representation does not perform well. Results of comparing IR1 and IR3 are

**Table 3** Experimental results for each of the parameters univariately. The sub-tables show the results of each of the EDA parameters individually. Each sub-table summarizes a total of 1440 runs, from which the best results are shown in bold. The results displayed are mean fitness ( $\bar{S}$ )  $\pm$  standard deviation ( $sd$ ), best fitness, and how many times best fitness was achieved

(a) EDA type											
EDA type	Bertin4		Bertin8		Bertin32		Bertin128		Hospitals		
	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	
discrete	506.43 $\pm$ 137.98	332 (106)	968.8 $\pm$ 394.1	476 (10)	5408.36 $\pm$ 3251.2	1304 (1)	40208.5 $\pm$ 8975.73	29116 (1)	<b>330.88 <math>\pm</math> 9.26</b>	300.725 (1)	
continuous	<b>402.41 <math>\pm</math> 57.17</b>	<b>330 (140)</b>	<b>822.66 <math>\pm</math> 369.3</b>	<b>474 (130)</b>	<b>4102.1 <math>\pm</math> 3479.6</b>	<b>978 (66)</b>	<b>18225.43 <math>\pm</math> 18378.17</b>	<b>1986 (38)</b>	344.98 $\pm$ 24.17	<b>299.27 (4)</b>	

(b) EDA complexity											
EDA complexity	Bertin4		Bertin8		Bertin32		Bertin128		Hospitals		
	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	
univariate	448.30 $\pm$ 121.27	<b>330 (37)</b>	910.63 $\pm$ 362.64	<b>474 (49)</b>	4666.90 $\pm$ 3353.68	<b>978 (25)</b>	30666.4 $\pm$ 18157.7	<b>1986 (24)</b>	338.68 $\pm$ 20.70	<b>299.27 (2)</b>	
bivariate	448.03 $\pm$ 96.75	<b>330 (48)</b>	<b>858.20 <math>\pm</math> 291.72</b>	<b>474 (48)</b>	<b>4420.5 <math>\pm</math> 3212.65</b>	<b>978 (35)</b>	30508 $\pm$ 18122.76	<b>1986 (14)</b>	<b>337.33 <math>\pm</math> 20.21</b>	<b>299.27 (2)</b>	
multivariate	<b>446.94 <math>\pm</math> 132.76</b>	<b>330 (25)</b>	917.73 $\pm$ 488.01	<b>474 (33)</b>	5178.29 $\pm$ 3700.53	<b>978 (6)</b>	<b>26475.68 <math>\pm</math> 18232.48</b>	1988 (38)	337.77 $\pm$ 18.08	302.22 (2)	

(c) Individual representation											
Individual representation	Bertin4		Bertin8		Bertin32		Bertin128		Hospitals		
	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	
IR1	483.49 $\pm$ 168.06	<b>330 (140)</b>	905.22 $\pm$ 450.38	<b>474 (130)</b>	6033.54 $\pm$ 4518.33	<b>978 (66)</b>	35890 $\pm$ 23822	<b>1986 (38)</b>	<b>320.31 <math>\pm</math> 12.5</b>	<b>299.27 (4)</b>	
IR2	442.08 $\pm$ 46.37	332 (58)	1032.33 $\pm$ 318.65	476 (3)	5071.6 $\pm$ 2213.75	1304 (1)	33380 $\pm$ 557.3	29856 (1)	347.1 $\pm$ 16.84	316.18 (2)	
IR3	<b>437.70 <math>\pm</math> 101.42</b>	332 (84)	<b>749.01 <math>\pm</math> 333.77</b>	476 (13)	<b>3160.5 <math>\pm</math> 2425.14</b>	980 (1)	<b>18381 <math>\pm</math> 15850</b>	1988 (6)	346.39 $\pm$ 15.77	316.81 (1)	

**Table 3** (Continued)

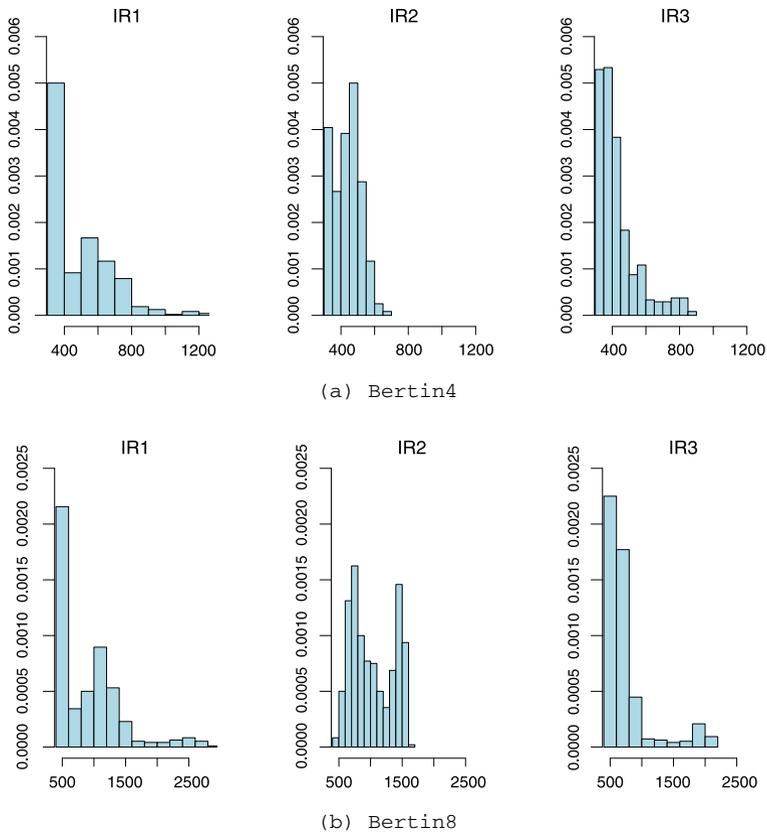
(d) <i>Elitism</i>		Bertin4		Bertin8		Bertin32		Bertin128		Hospitals	
	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$
yes	<b>426.96</b> ± <b>85.78</b>	<b>330</b> (60)	<b>871.32</b> ± <b>322.31</b>	<b>474</b> (49)	<b>4932.96</b> ± <b>3541.6</b>	<b>978</b> (4)	<b>31988</b> ± <b>17539</b>	<b>1988</b> (18)	<b>334.36</b> ± <b>19.58</b>	<b>299.27</b> (2)	<b>334.36</b> ± <b>19.58</b>
no	481.88 ± 137.5	<b>330</b> (80)	919.73 ± 444.29	<b>474</b> (81)	<b>4577.5</b> ± <b>3307</b>	<b>978</b> (62)	<b>26445</b> ± <b>18444</b>	<b>1986</b> (38)	341.5 ± 19.02	<b>299.27</b> (2)	341.5 ± 19.02

(e) <i>Population size</i>		Bertin4		Bertin8		Bertin32		Bertin128		Hospitals	
	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$
$\alpha RC$	464.91 ± 129.25	<b>330</b> (60)	958.63 ± 446.66	<b>474</b> (54)	<b>4353.84</b> ± <b>3102.15</b>	<b>978</b> (31)	<b>29180.04</b> ± <b>18225.97</b>	<b>1986</b> (17)	339.55 ± 19.19	<b>299.27</b> (1)	339.55 ± 19.19
$2\alpha RC$	<b>443.94</b> ± <b>104.24</b>	<b>330</b> (80)	<b>832.41</b> ± <b>308.04</b>	<b>474</b> (76)	5156.62 ± 3686.9	<b>978</b> (35)	29253.89 ± 18197.88	<b>1986</b> (21)	<b>336.31</b> ± <b>19.93</b>	<b>299.27</b> (3)	<b>336.31</b> ± <b>19.93</b>

(f) <i>Selection size</i>		Bertin4		Bertin8		Bertin32		Bertin128		Hospitals	
	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$	Best (times)	$\bar{S} \pm sd$
Pop . Size/2	468.22 ± 141.89	<b>330</b> (72)	939.16 ± 422.10	<b>474</b> (63)	5100.78 ± 3693.18	<b>978</b> (37)	29329.4 ± 18161	<b>1986</b> (21)	339.11 ± 20.35	<b>299.27</b> (1)	339.11 ± 20.35
Pop . Size/4	<b>440.62</b> ± <b>85.30</b>	<b>330</b> (68)	<b>851.88</b> ± <b>347.05</b>	<b>474</b> (67)	<b>4409.68</b> ± <b>3108.31</b>	<b>978</b> (29)	<b>29104.5</b> ± <b>18262.1</b>	<b>1986</b> (17)	<b>336.75</b> ± <b>18.81</b>	<b>299.27</b> (3)	<b>336.75</b> ± <b>18.81</b>



**Fig. 6** Histograms of the stress measure for the four Bertin examples using the *Individual representation* parameter

more ambiguous, although the results for IR1 appear to be more disperse than IR3 as problem complexity increases, as detailed numerically in Table 3. On the other hand, Fig. 7 confirms that continuous EDAs perform much better than discrete ones, but differences among EDA complexity types are inconclusive (only the Bertin128 example is shown).

#### 4.4 Multivariate analysis of experiments

The previous section described a study univariately analysing the effect and influence of each parameter in order to measure the contribution of each parameter to the best fitness value. However, a multivariate study can reveal how each parameter influences the final result taking into account joint parameter variations.

The study in this section was applied to the toughest example, the Bertin128 table. Performance differences for this table were high in all the experiments. The results ranged from the fittest stress value of 1986 to the worst of 53930.

We propose the use of a Bayesian classification model that is capable of analysing and illustrating how different combinations of parameter values can influence the

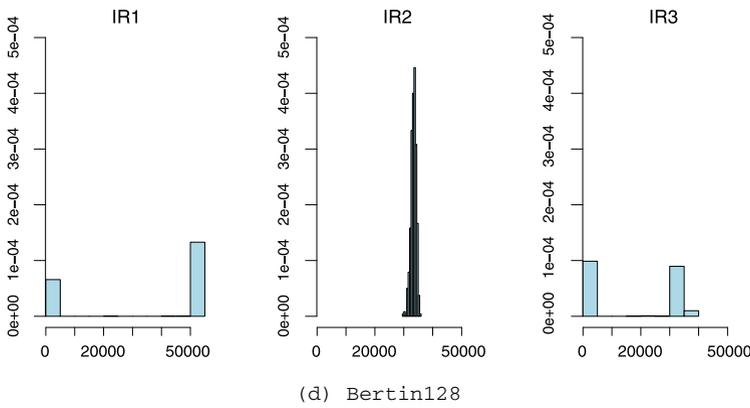
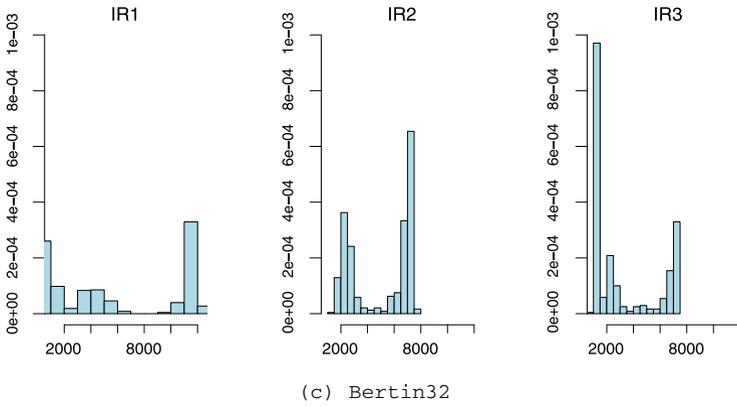
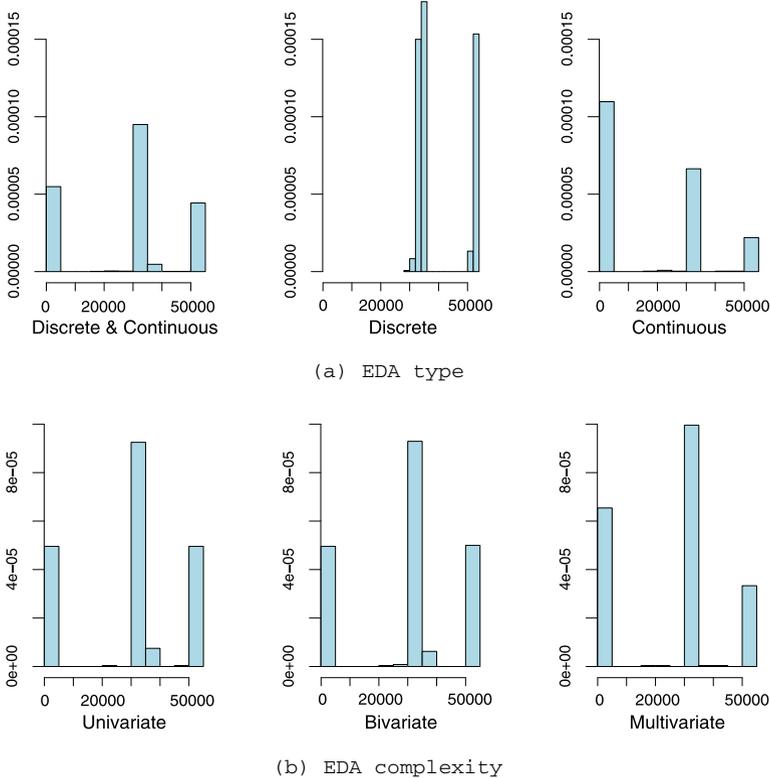


Fig. 6 (Continued)

overall performance of an EDA execution. This Bayesian classification model, called selective naive Bayes (Langley and Sage 1994), is described in the next section.

Bayesian classifiers are characterized by applying supervised classification approaches to concrete problems. The main goal of any supervised classification algorithm is to build a classification model using a data set. Our particular data set is the database of all the mean stress executions for the different parameter settings described in the previous sections. All these database parameters are considered predictor features. On the other hand, it is necessary in supervised classification to separate the cases into different classes. For the table ordering problem, we decided to order all the mean stress results for each combination of EDA parameters, considering the best 20% to be part of the class of best values (in Bertin128 this is any execution having obtained a mean stress value  $\bar{S} \leq 2555$ ), and the rest to be part of the class of worst values. The Bayesian classifier learning algorithm will analyse and illustrate which are the parameters (predictors) that influence the final result—by removing irrelevant or redundant parameters—to determine when an execution will finish the search within the range of the best 20% runs. More formally, this is expressed by



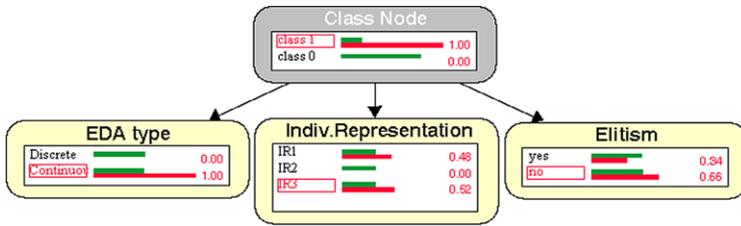
**Fig. 7** Histograms of the stress measure for the Bertin128 example using the different *EDA type* and *EDA complexity* parameters

a new variable added to the model, the *class variable*, which will take two possible values: 1 for the class of best runs, and 0 for the class of worst runs.

#### 4.4.1 The selective naive Bayes classifier

The *naive Bayes* classifier (Minsky 1961) is the simplest possible Bayesian classifier characterized by assuming that all predictor variables are conditionally independent given the class variable. This classifier is robust with respect to irrelevant variables, but is very sensitive to redundant variables (Inza et al. 2000). As a result, redundant variables decrease the accuracy of naive Bayes classifiers (Langley and Sage 1994). For this reason, a feature subset selection process (FSS) (Kohavi and John 1997) is usually a good option for removing those variables and obtaining a new subset of predictor variables to induce a more efficient classifier.

The *selective naive Bayes* classifier (Langley and Sage 1994) is a combination of FSS and the naive Bayes classifier. The main difference between the *selective naive Bayes* approach and naive Bayes is that selective naive Bayes can discard some of the predictor variables. Furthermore, the need to build a structure for the Bayesian classifier is an additional step not present in the naive Bayes model. This can be



**Fig. 8** Selective naive Bayes classifier learned using the database of executions for Bertin128

done in two standard ways: (a) forwards, starting with an empty set of variables and adding variables one by one, or (b) backwards, by removing one of the variables to be discarded at each iteration.

4.4.2 Applying the selective naive Bayes model to the EDA executions database

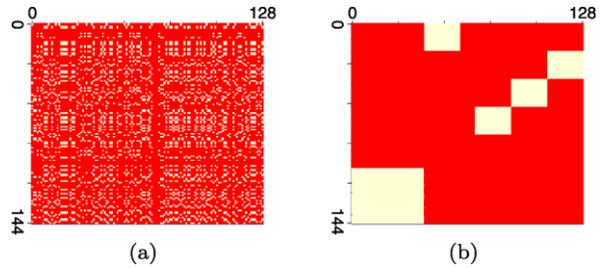
We learned the selective naive Bayes model structure in order to illustrate which parameters are relevant for providing a more efficient execution according to all the variations of EDA parameter settings. The model was learned from the database of all the mean stress results for all the parameter combinations described in Table 2, assigning for each case its class label (i.e. 1 or 0 as explained above). The learning algorithm applied was a greedy forward selective wrapper (Kohavi and John 1997) where probabilities are estimated using the maximum likelihood principle and Elvira software (Consortium 2002).

Figure 8 (top bars) shows the marginal probability distribution of each predictor for obtaining a stress result between 1986 and 2555, which corresponds to class 1. Only three of the six parameters are retained as relevant predictors: *EDA type* (discrete or continuous), *Individual Representation* (IR1, IR2 or IR3), and *Elitism* (yes or no). On the basis of the probability distributions learned by the classifier, it is possible to determine changes in the a posteriori probabilities for each predictor (EDA parameter) by propagating some evidence to the other classifier structure nodes. If we introduce evidence that they belong to class 1, our model gives probability 1 that the *EDA type* is continuous, and probability 0.52 that the *Individual Representation* is IR3 and 0.48 for IR1, see Fig. 8 (second bars). IR2 representation is never chosen. Regarding the *Elitism* parameter, the best results are obtained without elitism with a probability of 0.66.

An abduction study could perhaps extract more powerful information from this model. Abduction is the process of arriving at the most plausible explanations for a sequence of observed facts (evidence).

Within probabilistic systems such as Bayesian classifiers, abduction focuses on a search for the joint configuration of the non-observed variable values that yields the highest probability (Pearl 1988). The best explanation is the one that maximizes the joint conditional probability of the unobserved variables given the evidence. In this study, abductive inference is used to determine which are the *K* most plausible explanations for obtaining a stress value within the range 1986 to 2555 in Bertin128. The results for the abduction process are shown in Table 4 (*K* = 3). The fourth column reflects the joint probability distribution for each explanation.

**Fig. 9** (a) An image of a table of similar size to `Bertin128`; (b) output provided by the EDA using parameters suggested by the selective naive Bayes model for `Bertin128`



**Table 4** Most probable configurations of EDA parameters for `Bertin128` for obtaining a stress value between 1986 and 2555. The other configurations have a probability very close to 0

Explanation			Results		
EDA type	Indiv. represent.	Elitism	Probability	$\bar{S} \pm sd$	Best stress (times)
Continuous	IR3	no	0.023288	$2633.58 \pm 428.95$	1988 (2)
Continuous	IR1	no	0.020293	$2135.68 \pm 122.42$	1986 (38)
Continuous	IR3	yes	0.006449	$2779.33 \pm 768.80$	1988 (4)

The best results are obtained with continuous EDAs combined with a non-elitist approach, as already underlined in the univariate study. Regarding the individual representation, although both IR1 and IR3 are present, the results show that the highest probability of obtaining a result within the best 20% executions is when IR3 is applied. These results complement the outcomes described in Table 3. In Table 3 IR3 and the elitist approach appeared to be the best combination, whereas this multivariate study shows that the best combination is IR3 without elitism. Also, IR1 obtained worse mean results than IR3 in the univariate study, but this multivariate study shows that IR1 can also get good results if combined with continuous EDAs and a non-elitist approach.

Taking advantage of the selective naive Bayes study, we can determine good joint configurations of EDA parameters without having to run all the possible parameter configurations (thus, saving a lot of computational time). This is applicable to tables of similar characteristics to `Bertin128`, using the selective naive Bayes model to decide how to parameterize an EDA to get good solutions. For instance, let us consider a table of the same size as `Bertin128` illustrated in Fig. 9(a) with stress  $S = 40842$ . Figure 9(b) illustrates the image resulting from applying EDAs parameterized with the second configuration in Table 4. Stress is now  $S = 2114$ . Note that this is a satisfactory solution that is readily obtained without having to run all the possible configurations of EDA parameters as we did for `Bertin128`. The value for the *EDA complexity*, *Population size* and *Selection size* parameters were chosen at random.<sup>1</sup>

<sup>1</sup>The program can be freely downloaded from <http://www.sc.ehu.es/acwbecae/OrderingOfTables> having by default the best combination of parameters according to Table 4.

## 5 Conclusions and future work

This paper proposes the application of EDAs to the table ordering problem, analyzing the effect of their different parameters. For this purpose, three individual representations have been studied, two of them proposed in this paper for the first time. The study was performed with tables used in earlier similar studies in the literature to test the behavior of GAs on the table ordering problem, and the results of EDAs for both best and mean executions obtained here improve those published results.

Two studies were conducted: a univariate and a multivariate based on a selective naive Bayes model that reduced the number of relevant parameters from six to only three. These studies proved that EDAs in continuous domains are the most appropriate for this problem. Also, the validity of the newly proposed IR3 individual representation approach has been proved experimentally. From the selective naive Bayes model we found out that a continuous EDA with the IR3 representation is a good EDA parametrization regardless of the value of the *Elitism* parameter, which also becomes irrelevant. However, if a continuous EDA is combined with IR1, then non-elitism is the required choice.

For the future, this study should be extended using bigger non-binary tables that could better show how IR3 behaves. Preliminary results are not as good as for IR1, and further work is required to analyze the reasons for this. In addition, other individual representations not based on path representations could be formulated and tested for non-binary tables.

Finally, other metaheuristics (tabu search, scatter search, ant colony ...) rather than EDAs could be tried.

**Acknowledgements** This work has been partially supported by grants from the following institutions: Basque Government (Saiotek and Research Group programs 2007–2012 with grant IT-242-07; Spanish Ministry of Science and Innovation (grants TIN2008-06815-C02-01 and -02, TIN2007-62626, and Consolider Ingenio 2010—CSD2007-00018); Carlos III Health Institute (COMBIOMED network in computational biomedicine), and Cajal Blue Brain project.

## References

- Banfield, R., Raferty, A.: Model-based Gaussian and non-Gaussian clustering. *Biometrics* **49**, 803–822 (1992)
- Bengoetxea, E., Larrañaga, P., Bloch, I., Perchant, A., Boeres, C.: Learning and simulation of Bayesian networks applied to inexact graph matching. *Pattern Recognit.* **35**(12), 2867–2880 (2002)
- Bertin, J.: *Graphics and Graphic Information Processing*. Walter de Gruyter, Berlin (1981)
- Bielza, C., Fernández, J., Larrañaga, P., Bengoetxea, E.: Multidimensional statistical analysis of the parameterization of a genetic algorithm for the optimal ordering of tables. *Expert Syst. Appl.* **48**(4)
- Cabrera, J., McDougall, A.: *Statistical Consulting*. Springer, New York (2002)
- Cesar, J., Bengoetxea, E., Bloch, I., Larrañaga, P.: Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Pattern Recognit.* **38**(11), 2099–2113 (2005)
- Consortium, E.: *Elvira: An environment for creating and using probabilistic graphical models*. In: *Proceedings of the 1st European Workshop on Probabilistic Graphical Models*, pp. 222–230. Cuenca, Spain (2002)
- de Bonet, J., Isbell, C., Viola, P.: *MIMIC: Finding optima by estimating probability densities*. In: Mozer, M.J.M., Petsche, T. (eds.), *Advances in Neural Information Processing Systems*, vol. 9, pp. 424–431. Cambridge, MA (1997)

- Etcheberria, R., Larrañaga, P.: Global optimization with Bayesian networks. In: Special Session on Distributions and Evolutionary Optimization, pp. 332–339. II Symposium on Artificial Intelligence, CIMA99, La Habana, Cuba (1999)
- Friedman, J., Rafsky, L.: Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Ann. Stat.* **7**, 679–717 (1979)
- Friendly, M.: Corramms: Exploratory displays for correlation matrices. *Am. Stat.* **56**(4), 316–324 (2002)
- García, C., Pérez, D., Campos, V., Martí, R.: Variable neighborhood search for the linear ordering problem. *Comput. Oper. Res.* **33**(12), 3549–3565 (2006)
- Gómez, M., Bielza, C.: Node deletion sequences in influence diagrams using genetic algorithms. *Stat. Comput.* **14**, 181–198 (2004)
- Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer, J., Kanal, L. (eds.) *Uncertainty in Artificial Intelligence*. vol. 2, pp. 149–163. North-Holland, Amsterdam (1988)
- Inza, I., Larrañaga, P., Etcheberria, R., Sierra, B.: Feature subset selection by Bayesian networks based optimization. *Artif. Intell.* **123**(1–3), 157–184 (2000)
- Kohavi, R., John, G.: Wrappers for feature subset selection. *Artif. Intell.* **97**(1–2), 273–324 (1997)
- Koschat, M.: A case for simple tables. *Am. Stat.* **59**(1), 31–40 (2005)
- Langlely, P., Sage, S.: Induction of selective Bayesian classifiers. In: *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pp. 399–406. Seattle, WA (1994)
- Larrañaga, P., Etcheberria, R., Lozano, J.A., Peña, J.M.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: *Proceedings of the Workshop in Optimization by Building and Using Probabilistic Models*, pp. 201–204. A Workshop within the 2000 Genetic and Evolutionary Computation Conference, GECCO 2000, Las Vegas, Nevada, USA (2000)
- Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevich, S.: Evolutionary algorithms for the travelling salesman problem: A review of representations and operators. *Artif. Intell. Rev.* **13**, 120–170 (1999)
- Larrañaga, P., Kuijpers, C.M.H., Murga, R.H., Yurramendi, Y.: Searching for the best ordering in the structure learning of Bayesian networks. *IEEE Trans. Syst. Man Cybern.* **41**(4), 487–493 (1996)
- Larrañaga, P., Lozano, J.A.: *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic, Amsterdam (2001)
- Liu, K., Feng, J., Young, S.: PowerMV: A software environment for molecular viewing, descriptor generation, data analysis and hit evaluation *J. Chem. Inf. Model.* **45**(2), 515–522 (2005)
- Miller, G.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev.* **62**, 81–97 (1956)
- Minsky, M.: Steps toward artificial intelligence. *Trans. Inst. Radio Eng.* **49**, 8–30 (1961)
- Mühlenbein, H.: The equation for response to selection and its use for prediction. *Evol. Comput.* **5**(3), 303–346 (1998)
- Niermann, S.: Optimizing the ordering of tables with evolutionary computation. *Am. Stat.* **59**(1), 41–46 (2005)
- Niermann, S.: Letters to the editor. *Am. Stat.* **59**, 354 (2005)
- Pearl, J.: *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Palo Alto (1988)
- Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* **7**(2), 461–464 (1978)
- Shachter, R., Kenley, C.: Gaussian influence diagrams. *Manag. Sci.* **35**(5), 527–550 (1989)
- Walker, H., Durost, W.: *Statistical Tables: Their Structure and Use*. Bureau of Publications, Teachers College, Columbia University, New York (1936)