Software Platform for the Statistical Validation of Structural Partitioning Methods

Eduardo Xamena[a,b,c,d], *Ana Gabriela Maguitman*[c], *Nélida Beatriz Brignole*[a,b]

[a] *Planta Piloto de Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET*

+54 2914861700 Camino La Carrindanga Km 7, 8000 Bahía Blanca, Argentina.

[b] *Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC-DCIC),*

[c] *Knowledge Management and Information Retrieval Research Group - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA).*

Departamento de Ciencias e Ingeniería de la Computación (DCIC)

Universidad Nacional del Sur (UNS), Av. Alem 1253, 8000, Bahía Blanca, Argentina.

[d] *Facultad de Ciencias Exactas - Universidad Nacional de Salta (UNSa) - Av. Bolivia 5150, Salta, Argentina.*

E-mail: dybrigno@criba.edu.ar

Abstract

A novel interactive software platform is proposed to support design and development for PSE. It has been implemented to address the statistical validation of structural partitioning methods. According to several features given as model parameters, it produces incidence matrices whose structural partitioning can lead to a more efficient resolution of such models. The global objective is to generate automatically an arbitrary number of incidence matrices, shaped on the basis of statistical parameters associated with real-world PSE models. Then, partitioning methods can be executed on the generated matrices. Computational results for several problem instances are reported. Realistic cases were chosen by increasing the model-complexity level: a standard distillation column and an ammonia synthesis plant. In particular, the Direct Method, the Extended Direct Method and the Improved Extended Direct Method (IEDM) were evaluated. In comparison, the IEDM exhibited statistically significant enhancements of efficiency values for the resolution of the corresponding models.

Keywords: Structural partitioning; Incidence matrix; Statistical validation; Software platform; Systems of equations.

1. Introduction

Systems of linear algebraic equations occur naturally in almost every aspect of applied mathematics, including chemical engineering and scientific computing. Pre-processing strategies have been studied in general terms to obtain block partitionings that can be applied to general linear systems of equations. In particular, attention has recently been paid to solving sparse linear systems with modern architectures, so that the solution process reduces to the solution of much smaller independent systems within very simple iterative schemes. Drummond et al. (2015) considered a block projection method for the solution of sparse linear systems of equations, where they used a hypergraph partitioning in order to decouple the blocks, thus reducing the number of iterations. They have stated that a good pre-processing strategy can minimize the ill-conditioning between the different blocks. A robust efficient implementation was presented. Their successful parallel results indicate that the block Cimmino computational scheme is apt to be run on heterogeneous multi-level parallel systems.

Since linear systems of equations have been widely examined, we might infer that they constitute a guarantee for successful solving. In contrast, solving non-linear systems of equations is more complex for the following reasons, according to Heath (2002):

"1. A much wider range of behaviour is possible, so that a theoretical analysis of the existence and number of solutions is much more complex.

2. There is no simple way, in general, to guarantee convergence to the correct solution or to bracket the solution to produce an absolutely safe method.

3. Computational overhead increases rapidly with the dimension of the problem."

Therefore, it can intuitively be beneficial to bundle nonlinearities into blocks, rather than let a non-linear equation to intrude upon an otherwise linear block. The presence of nonlinearities in a subsystem forces solving the whole block by means of a non-linear technique, which seems to be less convenient.

Moreover, Bomhoff et al. (2012) have remarked that the effort required to find feasible solutions of a large system of equations grow as its size increases in the number of variables and equations. Therefore, it can be beneficial to decompose it into smaller subsystems. This decomposition is closely related to crown structures (Abu-Khzam et al., 2007) and it is carried out by means of solving the Free Square Block problem, where the algorithm can find the smallest subsystem to be solved separately. For some specific classes of graphs, crown reductions have also been investigated (Chlebík and Chlebíková, 2008).

The computation of big equation systems and the classification of their variables sometimes become much easier when structural partitioning is applied as pre-processing strategy. In the Process Systems Engineering (PSE) world, mathematical models with numerous equations often arise. The generated equations constitute systems that are often nonlinear. In the fields of simulation, optimization and instrumentation design of process plants, the incidence matrices derived from the associated systems of equations are generally sparse. For example, in Bike (1985) two models are described, which are represented by means of nonlinear systems of equations related to sparse incidence matrices. This scenario also emerges over various scientific areas, such as the modelling of physical systems (Nilsson et al., 2007) and Differential-Algebraic Equations

Systems (Tjoa & Biegler, 1991). Those models might benefit from partitioning techniques. The associated sparse matrices have been addressed in distinct ways. For example, Ponzoni et al. (2004) and Domancich et al. (2009) have applied their structure to PSE problems. By means of graph representations, the concomitant sparse matrices can be permuted to different structures, leading to a more efficient resolution of the associated model, or even to the determination of a greater amount of variables by the resolution of the systems.

Some graph-based algorithms, which will briefly be summarized in Section 2.1, are useful for the attainment of the desired permutation on matrices. Their structural objective is to generate a Lower Block-Triangular Form (LBTF) (Duff & Reid, 1979), where square assignment blocks can be distinguished on the main diagonal of the matrix. Those blocks represent small subsystems of equations, which can be employed to compute the values of the involved variables by means of the corresponding equations. With these smaller blocks, it would not be necessary to solve the system of equations as a whole. Instead, the blocks can be solved as independent subsystems, sequentially or in parallel, depending on the coupling of the variables among blocks. Moreover, in case there are linear blocks, they will be detected. Favouring this ordering, as proposed in this paper, makes it generally easier to solve large systems of linear equations efficiently because linear blocks are isolated. Given an LBTF, the amount of determinable variables will depend on the size and amount of the assignment blocks. This computation can be implemented with the help of structural partitioning methods.

In the field of instrumentation design of process plants LBTF is a useful pattern to carry out the observability analysis, which determines the amount of information that can be obtained from the available instruments by means of the corresponding model of the process in steady state (Ponzoni et al., 2004). Moreover, in simulation and optimization (Cucek et al., 2011) the resolution of large systems of equations can be more efficient through acceleration with an appropriate use of the assignment blocks generated by partitioning methods.

When solving a system of equations, it may be convenient not to solve it entirely at once. Instead, it is advisable to decompose it with an adequate LBTF into smaller subsystems that can be solved in order. Incidence matrices can be permuted to an LBTF pattern by means of the structural partitioning technique called the Direct Method (DM) (Ponzoni et al., 2004), which applies graph theory and algorithms for the attainment of an LBTF. Later, the Extended Direct Method (EDM) (Domancich et al., 2009) was developed on the basis of the DM. It includes a calculation procedure of the complexity of equations and variables in the system that privileges the generation of blocks of linear equations. Finally, the latest evolution of the DM is the Improved EDM (IEDM) (Xamena et al., 2012). This enhancement on the EDM performs a prior ordering on the equations by their complexity degree. In terms of the simplicity of the obtained assignment blocks, the IEDM overcomes the other two methods in some cases.

One of our main goals is to bring together these three approaches and highlight the way they work in the development of efficient algorithms. In particular, for mathematical models to be executed in a widely used software package, like GAMS (Brooke et al., 1992), it is advisable to test whether the incorporation of a pre-processing with a partitioning method looks justifiable. A useful way of testing the apparent enhancements is an empirical, statistical validation over a considerable quantity of PSE cases. In this sense, a beneficial line of investigation presented in this paper is the development of a tool that generates case studies automatically, according to parameters associated with real problems and theoretical plants. Our software platform performs the generation of several incidence matrices and other data structures related to process plants. All these data are built taking into account parameters that reflect some characteristic features of the respective cases. For example, a parameter could be the proportion of linear equations that makes up the system of equations associated to a plant.

In this paper the algorithm employed by the structural partitioning methods corresponds to the original version of Tarjan´s algorithm (Tarjan, 1972), which identifies strongly connected components (SCC) of a directed graph by making a Depth First Search (DFS). Duff (1977) surveyed the state of the art in sparse matrix research. According to this paper, two of the major algorithms for a structural approach in this area were developed by Tarjan (1972) and Sargent and Westerberg (1964), whose implementations were described in Duff and Reid (1978). Other implementations, which were reviewed in Lowe (2015), have been generated for SCC identification on the basis of these algorithms for parallel architectures. For instance, Fleischer et al. (2000) proposed a divide-and-conquer algorithm for the same purpose, where the main difference between the original sequential approach and Fleischer´s proposal is the absence of a DFS mechanism.

In the DM (Ponzoni et al, 2004) Tarjan´s method (Tarjan, 1972) had been chosen for its fine decomposition stage. Nowadays, it is still a valid relevant choice to be kept since the concurrent implementation of Tarjan´s algorithm has recently been addressed (Lowe, 2015) in detail. Therefore, it can be inferred that it is possible to take advantage of parallel computing by adapting DM´s algorithm, and consequently IEDM´s algorithm, to achieve better performance in new architectures.

In this work a software platform for the statistical validation of structural partitioning methods for PSE models is presented. Section 2 pinpoints the most relevant details about the Structural Partitioning Methods that should be kept in mind. Then, in Section 3 the features of the new platform that allows the generation of case-studies, together with interesting statistical results about the corresponding partitioning method, are described, also including the core algorithms as well as the choice and handling of parameters. The subsequent section describes some results and some implementation alternatives for the platform are discussed. Finally, the last section refers to the main conclusions.

2. About Structural Partitioning Methods

The structural partitioning methods chosen for the platform are those that attain a suitable shape of incidence matrices, in the sense that it turns the resolution of the related systems of equations more efficient. Based on the graph representations of the corresponding matrices, the blocks are found in two steps by means of the consecutive execution of two algorithms for Coarse- and Fine-grain Decompositions, respectively. In the next subsections, several partitioning methods mentioned in the Introduction are briefly described.

2.1. Overview

The DM, EDM and IEDM perform permutations over incidence matrices with the purpose of arriving to an LBTF. The first step in this process is the *Coarse Decomposition* (Ponzoni et al., 2004). The procedure consists on building a bipartite graph or *bigraph*, from the rows, columns and relations of an incidence matrix, and then obtaining a maximum matching on that structure, as shown on Fig. 1. An incidence matrix **A** with 8 rows $R = (r_1, r_2, \ldots, r_8)$ and 7 columns $C = (c_1, c_2, \ldots, c_7)$ has been considered to build this example.

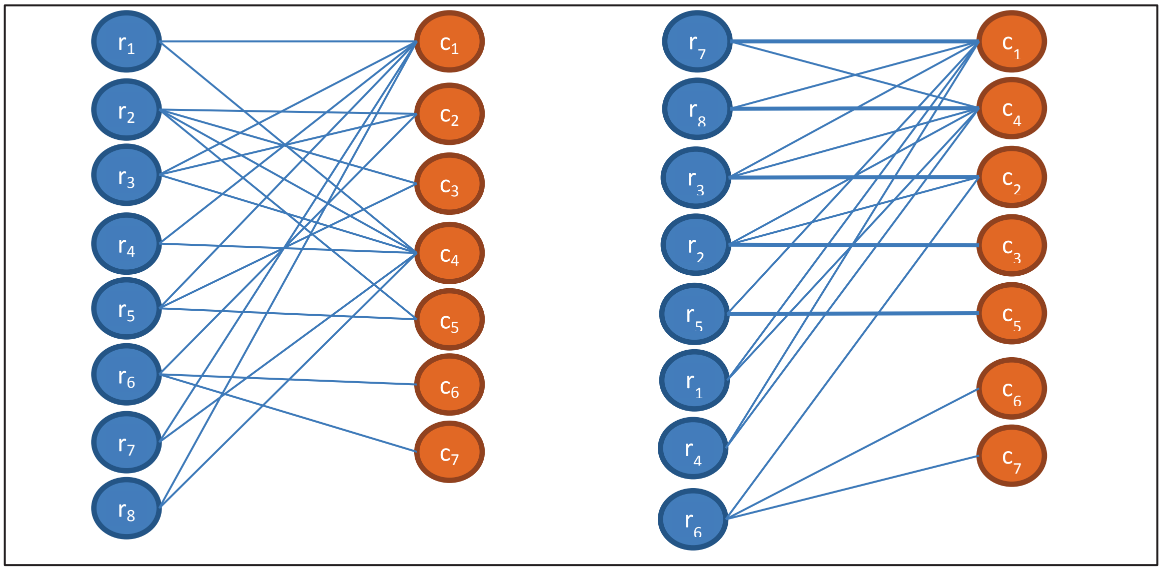


Fig. 1. A bipartite graph (left) and a related maximum matching (right).

In this bipartite matching, every edge $E_{i,j} = (r_i, c_j)$ that connects row r_i with column c_j is associated with **A**'s element $\alpha_{ij} = 1$. The algorithm described in Hopcroft and Karp (1973) is employed for the attainment of a maximum matching on this bipartite graph.

The bipartite graph representation is closely related to the solvability of systems of equations. Let us define the determinable (or observable) variables as those whose values can be found through the system of equations, while the indeterminable (or unobservable) variables appear whenever the system is under-specified. After computing the matchings, the bigraph can be split into seven sets: SR1, SR2, SC1, SC2, VR, HR and HC, as shown in Fig. 2.

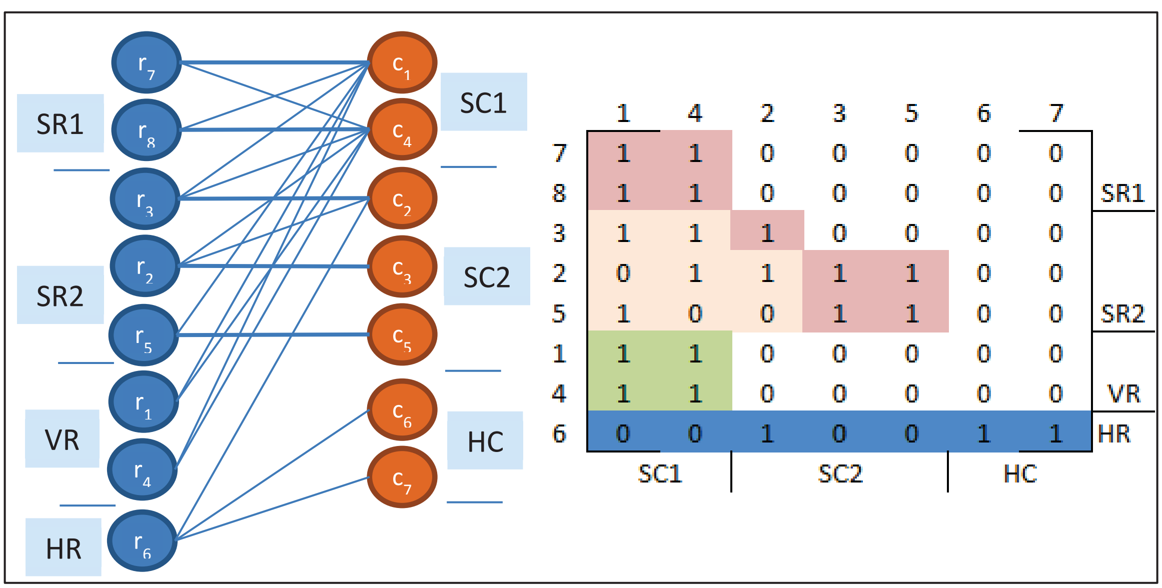


Fig. 2. Partitioning that arises from a maximum matching.

Every one of these sets comprises a different kind of nodes, according to this classification:

- SR1-SC1: The first group of determinable variables and assigned equations. These equations can be replaced eventually by the ones in the VR-SC1 group.
- SR2-SC2: The second group of determinable variables. The peculiarity of this group lies in the fact that its equations cannot be replaced by equations that belong to the VR-SC1 group.
- VR-SC1: Equations that only contain variables of the SC1 group. These equations were not included in SR1-SC1, but eventually might be introduced in that group, alternating them with other equations.
- HR-HC: Equations with indeterminable variables.

As to the incidental overlapping of SR1-SC1 and VR-SC1, the group of equations that will be used for the determination of variables in the SC1-SC2 group is the SR1-SR2 subset. At the resolution stage, only the equations in SR1-SR2 are used. The purpose of the equations´ block VR is to keep further equations that only have incident variables of the block SC1, and it would be considered in case of finding forbidden blocks during the rearrangement procedure, to replace SR1 equations. In fact, the distinction between groups SR1-SC1 and SR2-SC2 is the absence of equations in VR for the eventual replacement of an equation of SR2, when a forbidden block is found in the set SR2-SC2. For this reason, the equations in VR are called "redundant equations", and are not employed to solve the system. Hence, the overlapping of VR-SC1 and SR1-SC1 corresponds to the redundancy taken into account by Ponzoni et al. (2004).

It should be noted that this partitioning process is structural in nature, but it serves to guide a procedure of numerical solving. In the square block SR1-SR2/SC1-SC2, the final LBTF structure is free from singularities because every singularity is considered as a forbidden block during the partitioning process. For example, for instrumentation design purposes singularities are set beforehand based on technical reasons. There are forbidden subsets (Ponzoni et al., 2004) to take into account foreseeable numerical problems that are going to arise when the corresponding subsystem has to be numerically solved. Then, restrictions are imposed whenever a subsystem composed of numerically singular equations appears (Domancich et al., 2009).

As to its dimension, the complete block SR1-SR2/SC1-SC2 is produced by the partitioning methods; thus, its shape is always square. Moreover, since it corresponds to a maximal matching of a bigraph (Hopcroft and Karp, 1973), the number of rows is equal to the number of columns. Hence, it is impossible to generate non-square SR1-SR2/SC1-SC2 blocks.

After having performed the maximum matching algorithm and recognized the described sets of nodes, the next step consists in building the assignment blocks of equations and variables inside SR1-SC1 and SR2-SC2. This task is called *Fine Decomposition* (Ponzoni et al., 2004). The assignment blocks correspond directly to the *Strongly Connected Components* (SCC) of each one of the determinable-variable blocks SR1-SC1 and SR2-SC2. The algorithm employed for this purpose is the one explained in Tarjan (1972). As an example, let us consider the subsystem shown in Eq. 1 resulting from **A**, as stated in Fig. 1 and Fig. 2.

$$\begin{cases} r_2: & f_4(x_2, x_3, x_4, x_5) = 0 \\ r_3: & f_3(x_1, x_2, x_4) \quad = 0 \\ r_5: & f_5(x_1, x_3, x_5) \quad = 0 \\ r_7: & f_1(x_1, x_4) \qquad = 0 \\ r_8: & f_2(x_1, x_4) \qquad = 0 \end{cases} \tag{1}$$

In Fig. 3, the SCC and the permuted matrix for this example of a group of assignment blocks can be observed. Fig. 3 illustrates how the partitioning methods analyzed in this paper (DM, EDM and IEDM) work in general. The different square blocks outlined here depict the assignment blocks that can be obtained by means of the application of the two main procedures of the partitioning methods: The Maximum Matching algorithm and the SCC Detection algorithm.

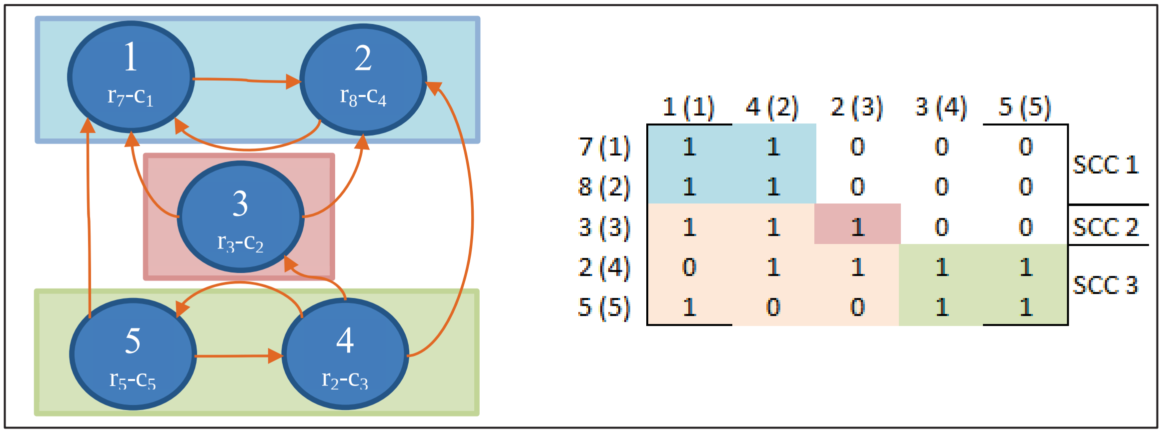


Fig. 3. Fine decomposition with SCC's and assignment blocks of an incidence matrix.

After the fine decomposition has been executed, the square blocks of the main diagonal of the resulting matrix correspond to the SCC found. Every SCC is directly mapped to an assignment block. Each one of these assignment blocks determines a subsystem of equations. Those subsystems can be solved as independent units. The only constraint for the process of solving the subsystems is the coupling of variables with prior blocks. It derives in the need of a sequential resolution. In other words, a subsystem cannot be solved unless the variables present in its equations that are not part of the associated assignment block have been already set. From the structure of the resulting matrix in Fig. 3, the sequence of resolution of assignment blocks can be stated. For the example given by Eq. 1, a complete solution can be obtained by first solving f_1 and f_2 as a subsystem, substituting the values found for x_1 and x_4 into f_3. Finally, after substitution of the values found for x_1, x_2 and x_4, the subsystem comprising f_4 and f_5 can be solved.

When efficient solvability is required, an important aspect of the DM is the absence of a mechanism for classifying variables and equations according to their complexity. This issue corresponds to the kind of every term of the equations. If an equation is only made up by linear terms, then it should be considered less complex than others; for example, than polynomials or those with derivatives. If a variable only appears in linear terms of equations, then it is less complex than any other variable that is present in nonlinear terms. By analysing straightforwardly the system of equations originated by **A**, the EDM (Domancich et al., 2009) performs such classifications, including this feature in the process of Coarse Decomposition. The linear equations and variables are ordered in the first place, thus leading to a configuration where it is easier and quicker to find a solution. In short, a better configuration of assignment blocks is achieved as a result.

Case Study	Amount of	DM	EDM	Δ
Distillation Column	observable variables	63	63	0
	unobservable variables	22	22	0
	1 × 1 subsystems	48	58	+21%
	linear subsystems	47	57	+21%
	nonlinear subsystems	4	2	-50%
Ammonia Synthesis Plant	observable variables	216	216	0
	unobservable variables	297	297	0
	1 × 1 subsystems	144	164	+14%
	linear subsystems	143	164	+15%
	nonlinear subsystems	8	4	-50%
Ethane Plant	observable variables	929	929	0
	unobservable variables	673	673	0
	1 × 1 subsystems	767	777	+1.3%
	linear subsystems	758	776	+2.3%
	nonlinear subsystems	26	20	-23%

Table 1. Performance differences between DM and EDM for some case-studies.

In Table 1 it can be noticed that EDM outperforms DM in terms of the quantity of linear blocks found for some examples extracted from (Domancich et al., 2009). This behaviour is due to the prior classification of variables and equations that EDM makes with the system of equations. The PSE cases whose results are here exhibited are representative of some advantages of pre-processing through EDM for the easy resolution of the attained subsystems. In the last column, Δ indicates the percentage of quality changes in the subsystems yielded by EDM with respect to those indicated by DM. The negative sign indicates a reduction in the amount of such subsystems.

In other words, Table 1 reports the sizes of the block decomposition stemming from the individual application of each method for three case studies. Aiming at a quality comparison between DM and EDM, the percentage Δ denotes the difference in the amount of blocks of a given category yielded by each method. For example, for the Ammonia Synthesis Plant, there is an increase of Δ=15% in the number of linear subsystems favouring the EDM over the DM, and the nonlinear subsystems decrease in a half, i.e. Δ=-50%, also favouring the EDM, given that the blocks are desired to be linear.

2.2. Improving structural partitioning by means of EDM and IEDM

The classification performed by the EDM employs a prior ordering on the set of equations and variables of the system. Firstly, an EDM routine computes the Non-linearity Degree (NLD) of all the equations and variables. After those NLD values are calculated, the EDM orders the adjacent nodes of every equation and variable in ascending order. The ordered vector of adjacent nodes constitutes the source of information that the algorithm employs to extract adjacent nodes whenever it is necessary to match a node.

The NLD of a given equation is exclusively envisaged from a linear algebra viewpoint. It is computed by considering the linearity type of each term in the equation. For instance, a term can be linear, bilinear, or nonlinear, depending on the mathematical operations that it involves. To calculate the NLD, a specific weight is assigned to each term according to its linearity type. The criterion presented and illustrated in Domancich et al. (2009) tends to favour the linear terms because smaller weights are assigned to the terms with lower NLD.

Moreover, the IEDM continues with an extra step that performs an additional ordering on the equations vector (Xamena et al., 2012). Due to this strategy, the number of linear assignment blocks might increase. Hence, the obtained partitioning can be better than the one yielded by a method that only performs the prior ordering on the adjacent nodes, in terms of the complexity of equations that make up every block.

Fig. 4 presents another example of a system of equations where IEDM attains better structural partitioning than EDM. Table 2 shows the location of the constraints in the formation of assignment blocks coming from this system. The constraints are sets of variables and equations that should be removed as assignment blocks in order to guarantee solvability. For the system in Fig. 4, 14 constraints of size 1 (listed on the left) and 3 sets of size 2 (listed on the right) were individualized. Table 2 reports the indexes (i,j) for the matchings (x_i,e_j). On the right, each band with 2 rows shows the matchings that constitute a set of size 2.

$$S_1: \begin{cases} e_1: x_2 + 3x_7 - 2x_{10} - 10 = 0 \\ e_2: \log_{10}(x_2^3 - 17) + x_2^2 - 10 = 0 \\ e_3: -3x_2 - 3x_8 + 15 = 0 \\ e_4: 3x_2 - x_7 - x_{10} = 0 \\ e_5: x_6 + x_{10}^2 + e^{x_{10}-4} - 14 = 0 \\ e_6: 7x_1 - x_5 + \dfrac{x_2^3}{3} + x_2^2 - 23 = 0 \\ e_7: 2x_2 - 3x_7 + 9 = 0 \\ e_8: x_6 + x_7 - 2 = 0 \\ e_9: 3x_2 - x_6 + x_7 + 2x_{10} - 25 = 0 \\ e_{10}: 2x_2 + x_6 - 2x_7 - x_{10} + 11 = 0 \\ e_{11}: -8x_1 + 15x_6 - 20x_7 - x_{10} + x_2^4 * x_8 + x_2^2 - 14 = 0 \\ e_{12}: x_2^3 - e^{(x_2-3)} - x_3 - 4x_4 - x_5 + x_6 - x_7 + x_9 - x_{10} + 4 = 0 \end{cases}$$

Fig. 4. Example of a system of equations built with the case generation platform.

If IEDM is applied on the structure of the system in Fig. 4, together with the constraints given in Table 2, the structural partitioning reported in Table 3 is obtained. In this reorganization, there are three linear blocks and two nonlinear blocks, individualizing 5 determinable variables (x_2, x_{10} , x_7 , x_6 and x_8) that can be computed in the first place through linear blocks and 2 determinable variables (x_1 and x_5)that are to be calculated afterwards by means of non-linear single equations.

	Variables	Equations		Variables	Equations
	10	8		9	4
	4	7		6	5
	9	12		2	3
	9	5		4	11
	8	12		4	4
	7	6		1	11
Size 1	3	8	**Size 2**		
	4	12			
	5	12			
	7	9			
	10	12			
	9	2			
	9	6			
	3	7			

Table 2. Constraints for assignment-block formation on the system in Fig. 4

In contrast, when the DM is executed, only one linear equation appears isolated in (x_8,e_3), while the rest of the determinable variables staying involved in non-linear blocks. Apart from individualizing this block, the plain EDM only yields another linear block: {(x_{10}, e_4); (x_7, e_1) }. Then, the improvements on IEDM can be appreciated in the higher amount of linear blocks obtained and hence, in the amount of variables easily computable by linear subsystems.

	x_2	x_{10}	x_7	x_6	x_8	x_1	x_5	x_3	x_4	x_9	Block Type	Group
e_7	1	0	1	0	0	0	0	0	0	0	LINEAR	SR1
e_4	1	1	1	0	0	0	0	0	0	0		
e_1	1	1	1	0	0	0	0	0	0	0		
e_{10}	1	1	1	1	0	0	0	0	0	0	LINEAR	
e_3	1	0	0	0	1	0	0	0	0	0	LINEAR	SR2
e_{11}	1	1	1	1	1	1	0	0	0	0	NONLINEAR	
e_6	1	0	0	0	0	1	1	0	0	0	NONLINEAR	
e_2	1	0	0	0	0	0	0	0	0	0		VR
e_5	0	1	0	1	0	0	0	0	0	0		
e_8	0	0	1	1	0	0	0	0	0	0		
e_9	1	1	1	1	0	0	0	0	0	0		
e_{12}	1	1	1	1	0	0	1	1	1	1		HR
Group	SC1				SC2			HC				

Table 3. Partitioning attained with the IEDM for Fig. 4

Table 4 summarizes the results of applying DM, EDM and IEDM to pre-process the structure of the system of equations in Fig. 4. It is also interesting to note that the amount of observable variables remains the same for the three methods. This is normal, because that value is the maximum possible amount of observable variables for this small example.

	DM	EDM	IEDM
Linear Blocks	1	2	3
Variables computable by linear blocks	1	3	5
Nonlinear Blocks	3	4	2
Variables computable by nonlinear blocks	6	4	2
Total number of blocks	4	6	5
Total number of observable variables	7	7	7

Table 4. Results of the system in Fig. 4 for DM, EDM and IEDM

3. Statistical validation platform

In PSE applications sparse systems of equations ensue from mathematical models, being decisive in simulation and optimization. By increasing the problem size or going closer to reality, the complexity and size of the underlying structure generally augments.

In this section a platform for empirical statistical validation is presented. A useful tool to generate case studies automatically is included. Such generation tool can be supplied with statistical parameters related to theoretical cases about real plants. The construction of this platform is described at the beginning. Then, the algorithms and data structures that make this platform work are explained.

3.1. The case generation platform

Building incidence matrices is the main issue of case generation to be able to test how the structural partitioning methods behave. Those matrices should be sparse and have a standard structure so that they reflect faithfully features of frequently employed mathematical models. Some relevant features are associated with the shape of the corresponding incidence matrices, their organization and other data structures. Since those matrices should be permuted to an LBTF, it is convenient to start with that kind of structure and shuffle it. For this purpose, a matrix is generated whose elements equal to one are assigned in random positions, according to statistical parameters which come from the selected mathematical models in the LBTF.

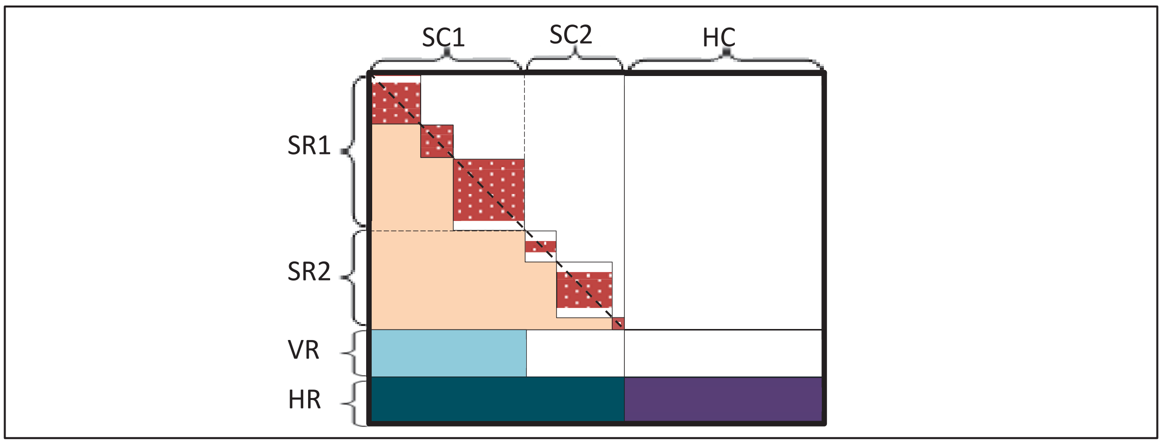


Fig. 5. LBTF of an incidence matrix, and the corresponding sets and assignment blocks.

The parameters that rule out the dynamic generation of an incidence matrix on the platform are derived from the size, density and position of each assignment block, as well as from the sets of variables and equations mentioned in section 2.1, i.e. SR1-SC1, SR2-SC2, VR-SC1 and HR-HC. Moreover, there are other noticeable model parameters that are related to the rectangles on the left of each assignment block and the rectangle on top of the indeterminable set HR-HC (see Fig. 5). Those blocks are also sparse, and it should be contemplated that for certain models they may have some non-zero entries. For an incidence matrix built with the platform, how a sparsity pattern looks is illustrated in Fig. 6. This pattern comes from a randomly generated matrix, originated by the case generation platform, where all the generated matrices are shaped according to the parameters of a reference case. The case-studies that exhibited this pattern are enumerated in Section 4.

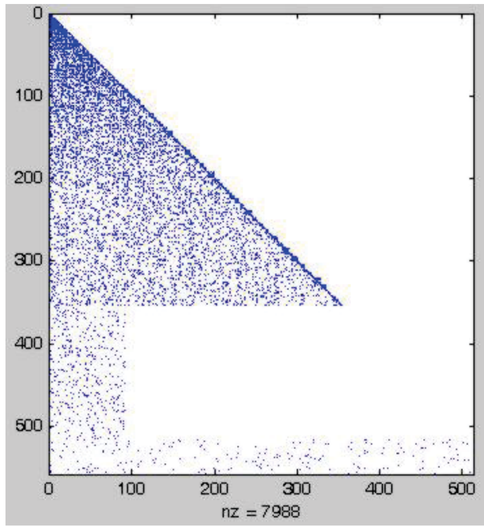


Fig. 6. Sparsity of an incidence matrix yielded by the generation platform.

Fig. 7 outlines the structure of the platform developed for the task of case-study generation. There are three modules inside that receive the model parameters and generate the corresponding data structures for a random case. The modules are the following:

1. Generator of incidence matrices
2. Generator of forbidden blocks
3. NLD generator for equations and variables

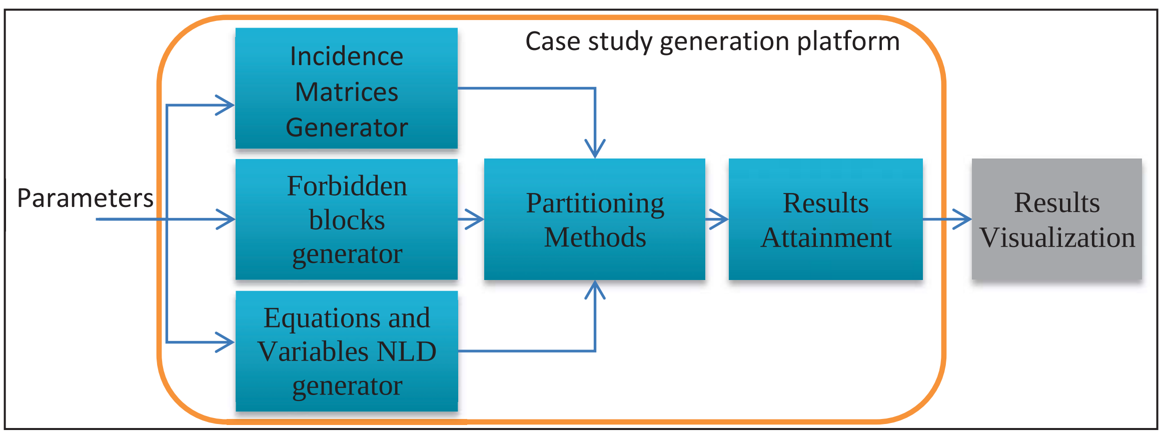


Fig. 7. Structure outline for the case study generation platform.

The first module elaborates the incidence matrix according to the parameters. The second one builds a data structure that includes several constraints on the formation of the assignment blocks for the partitioning methods. Finally, the third one produces the NLD for every variable and equation, according to the given parameters. For the generation of the incidence matrices, the parameters to be established, which are illustrated in Fig. 8, are the following:

1. Number of equations and variables in the system: These parameters determine the basic shape of the system of equations. The associated matrix can be chosen to be square or rectangular.
2. Size of the complete group SR1-SR2/SC1-SC2: it defines the number of observable variables and assigned equations of the corresponding square block. This is a random number, whose probability distribution function can be the Normal Distribution, with the mean and standard deviation provided by the user.
3. Size of the SR1-SC1 group: This quantity is lower than or equal to the priorly defined size of the order of the SR1-SR2/SC1-SC2 block. It is also normally distributed and set at random.
4. Size of the SR2-SC2 group: This is simply the difference between the orders of the SR1-SR2/SC1-SC2 and SR1-SC1 blocks.
5. Size of the assignment blocks inside SR1-SR2/SC1-SC2: The size of each assignment block is set according to random proportion values, also being normally distributed.
6. Size of the Block of Redundant Equations: The dimensions of this block depend on the number of equations of the remaining sets, and the size of the set SC1. The number of rows will be the difference between the total number of equations and the sum of the equations comprising the sets SR1, SR2 and HR.
7. Size of the Block of Indeterminable Variables: This value is established at random, on condition that in general terms there should be fewer equations than the number of indeterminable variables.
8. Number of variables that are present in every equation: This value determines the number of no-zero entries in each row of the generated matrix. Each row corresponds to an equation. In the case of group SR1- SR2/SC1-SC2, for each row this number is the sum of the non-zero entries in SC1 plus the quantity of non-zero entries comprising SC2. Instead of being normally distributed, this number is uniformly distributed because of its distinct behaviour.
9. Position of each variable in the corresponding row: Uniformly distributed integer value that establishes the column where each non-zero entry is located. It is calculated once for each non-zero entry.

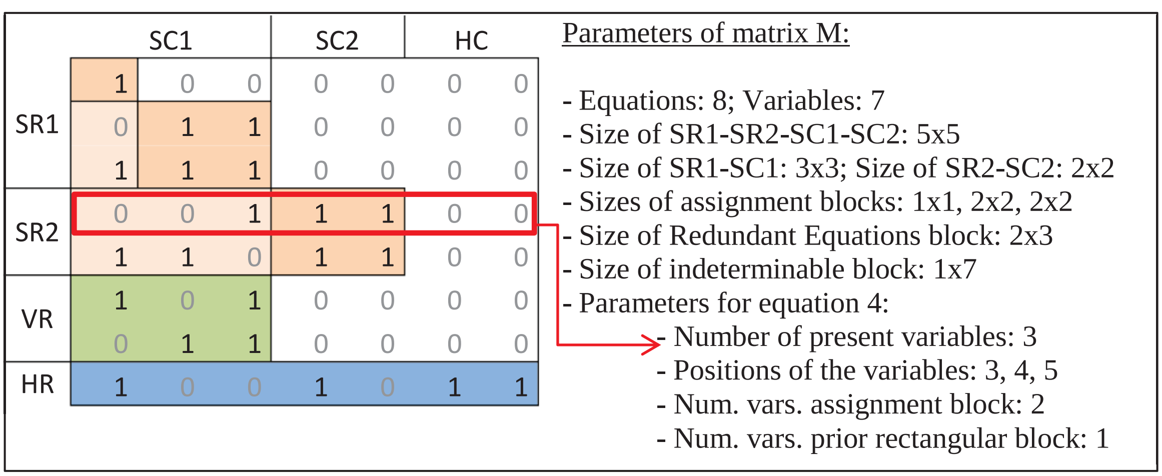


Fig. 8. Individualization of the parameters in the case-generation platform.

There are two additional parameters for the generation of case studies that allow taking into account some peculiarities of the mathematical models and mimicking them in the frame. The parameters that were included are the following:

- Number of forbidden blocks: This random value indicates how many forbidden blocks (constraints) will appear in the system of equations represented by the incidence matrix. These blocks are unacceptable since solvability cannot be ensured when they are present in the final partitioning.
- NLD for equations and variables: They are real numbers, also randomly defined, that establish the degree of nonlinearity associated with each equation and its variables. The definition and quantification of NLD are explained in detail in Domancich et al. (2009).

After the generation of a representative incidence matrix, the related constraints and NLD values, several partitioning methods are applied to the generated data. The output is summarized and processed in the *Results Attainment* module. Then, the *Results Visualization* module provides an appropriate display of the attained outcomes.

3.2. The algorithms for case generation

The functional steps of the developed platform should include the incidence matrices and their associated data structures and the generation of constraints, the application of the structural partitioning methods and the summary of the results. Based on the parameters that allow building the incidence matrices described in Section 2, this process was organized to gather statistical information about numerous representative cases developed by means of the platform. The complete algorithm for the generation and evaluation of test cases is reported in Fig. 9.

```
1.  Input: Parameters for incidence matrices building, Q: Number of cases to build
2.  Repeat Q times:
2a.     Generate randomly a matrix M in LBTF, based on the input parameters
2b.     Randomly reorder M rows and columns
2c.     Generate a set R of constraints for the assignment blocks formation, based on
        input parameters
2d.     Define random NLD for equations and variables
2e.     NLD=0 for a random group of equations and variables, according to the input
        parameters
2f.     Run DM, EDM and IEDM for the incidence matrix and the data structures
        attained
2g.     Save partitioning methods results
```

Fig. 9. Algorithm 1: Steps for the generation and evaluation of test cases.

Among the parameters fed to the program, some values correspond to the dimension of the represented systems, the average means μ and the standard deviations σ of the distributions that rule the sizes of the matrix sets, like the SR1-SC1 block size. By means of the regulation of such parameters, significant samples can be generated for separate matrix configurations.

Step 2 of Algorithm 1 (see Fig. 9) enters a loop cycle that generates ***Q*** case-studies, with ***Q*** given as input parameter. The first step inside this cycle builds a matrix **M** in LBTF, with the procedure described in Algorithm 2 (see Fig. 10). Next, in Step 2b, a random permutation on **M** is performed in order to give the partitioning methods a chance to find an alternative LBTF, without prior information of the initial LBTF.

The constraints for the formation of assignment blocks are assembled in Step 2c, and the maximum number of those constraints is also an input parameter. The construction of every constraint consists in setting random groups of equations and variables that do not take part of an assignment block. Besides, each constraint has a predefined size, indicating the amount of equations and variables it contains. For example, if the size of a constraint is 2, then it will be made up of 2 equations and 2 variables.

In Step 2d of Algorithm 1, the NLD of every equation and variable is randomly defined within the real interval between 0 and 3. An alternative way to obtain these values is the random generation of the terms of every equation in the system that will be represented; then, the NLD are calculated based on that structure. In Step 2e, a random group of equations and variables is set to be linear, i.e. their NLD will be equal to 0. It corresponds to the average proportion of linear equations of the original model, and there is a subset of input parameters related to this aspect of the platform.

```
1.  Input: Params (Statistical parameters for matrix building)
2.  sizeBlockSR1SR2=RandomInteger(Params->SR1SR2)
3.  sizeBlockSR1= RandomInteger (Params->SR1)
4.  sizeBlockSR2= sizeBlockSR1SR2- sizeBlockSR1
5.  Diagonal(BlockSR1SR2)=1

            Assignment Blocks Definition
6.  Repeat
6a.     Randomly generate a size for an assignment block
7.  Until SR1-SC1 is complete
8.  Repeat
8a.     Randomly generate a size for an assignment block
9.  Until SR2-SC2 is complete
10. For each assignment block in SR1-SC1 and SR2-SC2
10a.    For each equation (row)
            10a.i. Set number of variables of the equation and their positions, considering
            block bounds

            Redundant Equations Block and Indeterminable Block
11. IndetBlockColumns=Params. Variables-sizeBlockSR1SR2
12. IndetBlockRows= RandomInteger (IndetBlockColumns -1)
13. RedunBlockColumns=sizeBlockSR1;
14. RedunBlockRows=Params.Equations-sizeBlockSR1SR2- IndetBlockRows;
15. For each equation (row) of Redundant Block
15a.    Set number of variables of the equation and their positions, considering block
        bounds
16. For each equation (row) of the Indeterminable Block
16a.    Set number of variables of the equation and their positions, considering that each
        equation must have at least 2 indeterminable variables
```

Fig. 10. Algorithm 2: Construction of an Incidence Matrix from specified parameters.

Having generated the incidence matrix, the forbidden blocks and the NLD values for all equations and variables, Step 2f specifies the execution of the structural partitioning methods over the complete dataset. Each one of these methods (DM, EDM and IEDM) is run separately, and the obtained results are fed to the platform to be summarized in Step 2g.

The procedure indicated in Step 2a of Algorithm 1 is described in depth in Algorithm 2. It builds an incidence matrix, according to the parameters given as input. Steps 2 and 3 of Algorithm 2 assign the corresponding sizes to the complete block SR1- SR2/SC1-SC2, and the sub-block SR1-SC1. The size of the remaining block SR2-SC2 is the difference between the sizes of block SR1-SR2/SC1-SC2 and block SR1-SC1. Step 5 fills the main diagonal of the generated block with non-zero entries.

The next stage in the algorithm is the determination of assignment blocks. Steps 6 to 9 set their sizes. Then, each equation that belongs to a block is built with the quantity and positions of the variables randomly set. This task (Step 10) is performed carefully so that the block bounds are always considered, i.e. when setting positions for the variables inside the rectangle that corresponds to the block.

In Algorithm 2 Steps 11 to 16 carry out the formation of redundant and indeterminable blocks. The size of an indeterminable block depends on the total number of variables in the system, the size of block SR1- SR2/SC1-SC2 and the number of indeterminable variables. The columns of this block are the difference between the number of variables and the size of block SR1- SR2/SC1-SC2. Its rows are randomly set with the upper bound of the number of indeterminable variables minus 1. As to the redundant blocks, the number of their columns will be equal to the columns of SR1-SC1, and their number of rows will be the amount of remaining equations after the dimensions of the other blocks have been defined (Step 14). Steps 15 and 16 build the last blocks according to the dimensions previously set. The only remarkable thing is that every equation of the indeterminable block must have at least 2 indeterminable variables, because if there is only one, a new assignment block would be formed. Hence, the involved variable would not be indeterminable.

3.3. Statistical parameters for PSE models

The platform provides cases for a systematic study basically covering statistical ideas and techniques to test the performance of structural partitioning methods. It is intended to be useful for experimental and theoretical analysis of PSE problems.

The availability of information related to several academic and real cases of simulation and optimization problems is a key prerequisite for this task. Such information consists in statistical parameters that allow the creation of statistical samples. Along this section, several statistical parameters, extracted from two representative cases of Chemical Engineering and Instrumentation Design, have been depicted. The following examples have been analysed:

1. Example I- Reactive Distillation Column: This problem is related to the steady-state mathematical model of a distillation column that separates ammonia from a stream composed by ammonia and water (Domancich et al., 2009).
2. Example II- Ammonia Synthesis Plant: This is the model of a complete Haber-Bosch ammonia synthesis plant (Bike, 1985).

The parameters related to the incidence matrices and the constraints describing the system of equations for each model are summarized in Table 5. The first column specifies which parameter is evaluated for each mathematical model; the remaining columns report their values in the reference cases. The first two parameters reflect the dimension of the respective system of equations, which corresponds to the size of the incidence matrix A. The third row can be calculated by multiplying the number of equations by the number of variables. Then, it reports the amount of elements A has. The fourth row establishes the amount of non-zero entries (ones) in A. Next, the density of A was computed by dividing the Number of ones by the Number of cells. In both cases A has a very low density, illustrating the need for the design of data structures that hold only the non-zero entries. Next, there are two rows that report the size and amount of forbidden blocks that should be avoided for the formation of the assignment blocks. Finally, the amount of linear equations in the system is reported as a percentage of the total number of equations.

Parameter	Distillation Column	Ammonia Plant
Equations	102	557
Variables	85	513
Number of Cells	8670	285741
Number of Ones	265	1991
Density	3.06%	0.70%
Number of forbidden blocks	29	104
Maximum Forbidden Block Size	10	21
Percentage of linear equations	60%	54%

Table 5. Resulting parameters from the analysed cases

4. Results

The following representative problems, which were extracted from COPS collection (Dolan, 2004), were adopted as test problems:

- Catalyst Mixing (Catmix): The optimal policy of how to mix two catalysts along the length of a tubular plug flow reactor involving several reactions is determined.
- Shape optimization of a cam (Camshape): The area of the valve opening for a rotation of a convex cam with constraints on both the curvature and the radius is maximized.
- Hang glider (Glider): The final horizontal position of a thermal updraft is maximized.
- Robot arm (Robot): The time taken for a robot arm to travel between two points is minimized.
- Goddard rocket (Rocket): Given the initial mass, the fuel mass, and the rocket´s drag characteristics, the final altitude of a vertically launched rocket using the thrust as a control is maximized.

Nº	Name	Eq. #	Total # of Variables	Density	Proportion of linear Equations
1	Feed-15	138	137	3,32%	24,64%
2	Feed-9	84	83	5,29%	26,19%
3	Chloride	32	33	10,23%	34,38%
4	Distillation	102	85	3,06%	60,78%
5	Gasnet	49	58	5,03%	8,16%
6	Catmix	201	227	1,98%	0,50%
7	PropGlycol	28	29	10,71%	42,86%
8	Heat-Exchangers	58	59	2,98%	89,66%
9	Camshape	100	135	1,51%	100,00%
10	Methanol	367	312	1,29%	44,41%
11	Glider	610	612	0,60%	8,69%
12	Robot	403	484	0,74%	49,88%
13	Rocket	503	525	0,74%	44,14%
14	MTBE-Column	1923	1817	0,22%	21,53%
15	HDA	639	704	0,44%	75,27%
16	Ammonia	557	513	0,70%	54,04%

Table 6. Summary of the main features of the test problems

Besides, in order to complement and diversify our collection of test problems, the following representative cases were chosen:

- Optimal Design of a Gas Transmission Network (Gasnet): Design of a gas pipeline for the transportation of a fixed amount of gas. (Edgar et al., 2001).
- Reactive distillation for methyl tert-butyl ether synthesis (MTBE-Col): Economic optimization of a reactive distillation column model (Jacobs and Krishna, 1993).
- Hydrodealkylation of Toluene (HDA): Alternative process units for toluene hydrodealkylation into benzene and methane in a chemical plant. (Qiu et al., 2003).
- Feed plate location for a binary distillation column (Feed-9 and Feed-15): Interaction assessment for various feed plate locations by means of the Dynamic Relative Magnitude Criterion (Bendib and Khelassi, 2006).
- Propilenglycol production (PropGlycol): Maximization of the obtained profit (Kheawhom and Hirao, 2002).
- Ethyl chloride production (Chloride): Economic optimization that considers the costs associated to the production process (Domancich et al., 2004).

- Methanol production (Methanol): Conceptual design for the production of synthesis gas, which is suitable for methanol production (Cañete et al., 2014).
- Heat Exchangers Network (Heat-Exchangers): The optimization of an industrial network of heat exchangers arranged in two parallel sectors (Domancich et al., 2004).
- Reactive Distillation Column (Distillation): The steady-state model of a reactive distillation column (Domancich et al., 2009).
- Ammonia Synthesis Plant (Ammonia): The model of a complete Haber-Bosch ammonia synthesis plant (Bike, 1985).

For each case study, Table 6 depicts the parameters associated to the respective incidence matrices.

4.1. Example I: Reactive Distillation Column Problem

The configuration of the incidence matrix that corresponds to the mathematical model of a reactive distillation column (see Table 6: Case 4) is reported in Table 5. These settings were employed for the first evaluation instance.

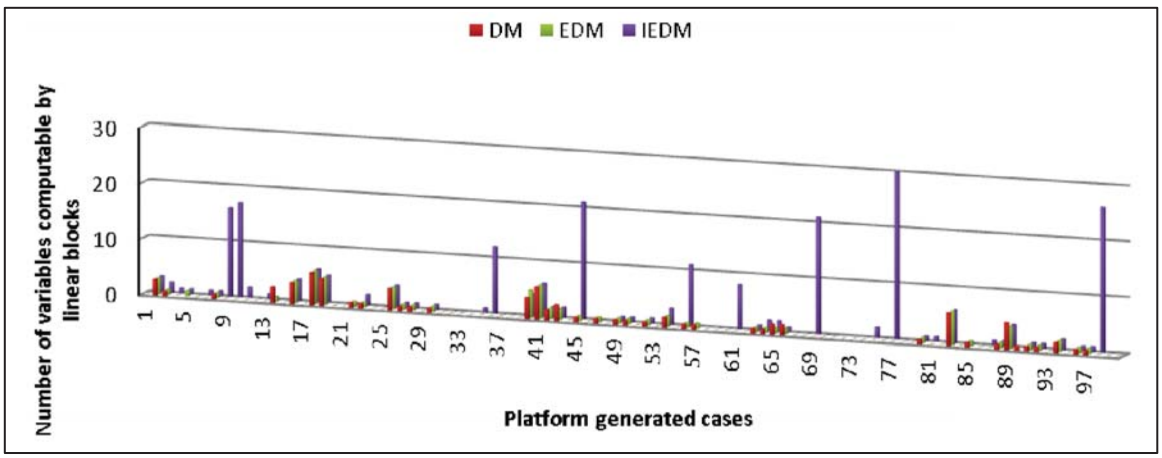


Fig. 11. Bar chart for 100 platform-generated cases for Example I

Fig. 11 displays the amount of variables computable by linear assignment blocks for 100 partitioned cases coming from Example I. This constitutes a representative sample of a larger population (1000 random cases) that was generated for this analysis. As to solvability, it can be concluded that the IEDM is more effective, yielding a more convenient partitioning than DM or EDM.

On Table 7 some statistical values corresponding to the cases considered in Fig. 11 are reported. The first column indicates which partitioning method was analysed. The second and third columns show the mean and standard deviation for the number of linear variables attained. The last column depicts the variability estimates through the calculation of the 95% Confidence Interval (CI). If non-overlapping of the CIs for DM and EDM against IEDM is considered, the IEDM looks the most convenient method, yielding a statistically higher amount of linear variables.

Partitioning Method	Mean	Std Dev	CI 95%
DM	0,83	1,450	0,546 - 1,114
EDM	0,78	1,411	0,503 - 1,057
IEDM	2,47	5,500	1,392 - 3,548

Table 7. Statistical results for the number of variables computable by linear blocks for 100 sample cases

As the number of observations gets larger, Table 8 ratifies the conclusion reached from Table 7. In Table 8, 1000 cases were analysed. The mean increased in more than 40% for the IEDM. Besides, the gap between the new CIs was even bigger. Hence, these results are again statistically favouring IEDM.

Partitioning Method	Mean	Std Dev	CI 95%
DM	1,17	2,148	1,041 - 1,307
EDM	1,19	2,213	1,055 - 1,329
IEDM	3,57	6,885	3,148 - 4,002

Table 8. Statistical results for the number of variables computable by linear blocks for 1000 sample cases

4.2. Example II: Ammonia Plant Problem

Fig. 12, Table 9 and Table 10 display the statistics taken for Example II (see Table 6: Case 16). The adopted configuration for the analysed population is listed in the column on the right in Table 5.

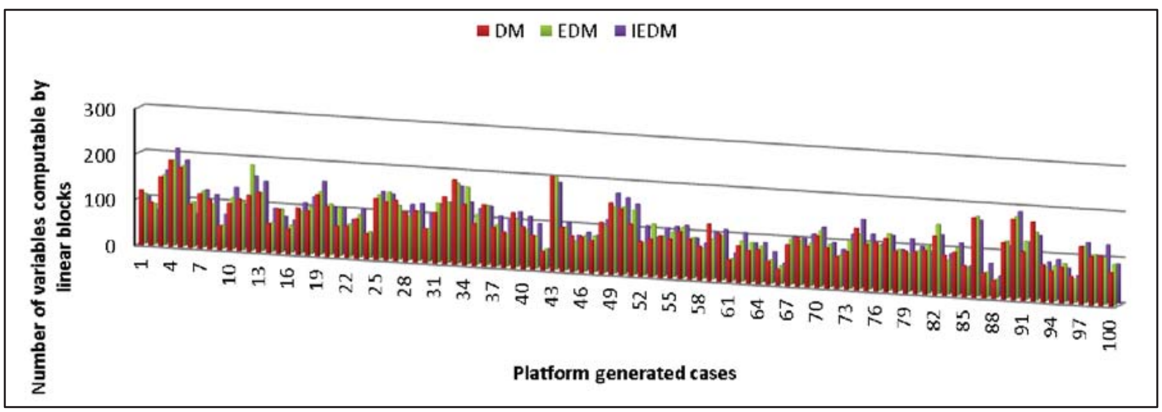


Fig. 12. Bar chart for 100 platform-generated cases for Example II

The statistical analysis for Example II was approached in the same way as the one adopted for Example I. In Fig. 12 a representative sample of 100 cases is displayed, which were collected from 1000 random cases of Example II. As to the attained quantity of linear variables, although the differences among methods are smaller in Fig. 12 than those in Fig. 5, there is still a considerable number of partitioning cases favourable to the IEDM in terms of the amount of linearly computable variables.

Partitioning Method	Mean	Std Dev	CI 95%
DM	103,54	34,137	96,84 - 110,23
EDM	103,80	35,126	96,92 - 110,68
IEDM	108,10	33,388	101,56 - 114,64

Table 9. Statistical results for number of linear variables for 100 cases on the configuration for Example II

On Table 9 the statistical values related to Fig. 12 are listed comprising for each method the following population parameters: the mean, the standard deviation and the 95% CI. In this sampling, there is an overlapping on the CIs, which is total for DM and EDM and partial in the interval for IEDM. These results do not reflect the statistical significance of the difference in the mean values. In this case, this sampling is not likely to account for the observed conclusions for Example I.

Partitioning Method	Mean	Std Dev	CI 95%
DM	100,14	32,428	98,13 - 102,15
EDM	100,03	32,141	98,04 - 102,02
IEDM	107,84	31,892	105,86 - 109,81

Table 10. Statistical results for number of linear variables for 1000 cases on the configuration for Example II

Nevertheless, on Table 10 encouraging values could be found. If the 1000 cases are considered, then the 95% CIs do not overlap when DM and EDM are taken as a group and contrasted with IEDM. Hence, the IEDM exhibits a statistically significant 7.7% enhancement on the amount of linear variables with respect to the other methods.

4.3. General Results

For each case study reported in Table 6, 1.000 instances of incidence matrices and NLD arrays for equations and variables were generated. All those structures were built by means of the generation platform in accordance with the parameters already stated for the problems. With this amount of information, the same statistical analysis described in Sections 4.1 and 4.2 were performed on the new cases. The results of those statistical studies are reported on Table 11 and Fig. 13. Table 11 enumerates the means and confidence intervals (CI) obtained for the number of variables computable by linear blocks in each case, and Fig. 13 shows a chart for these mean values and the respective CIs.

Nº	DM Lower Limit	DM Mean	DM Upper Limit	EDM Lower Limit	EDM Mean	EDM Upper Limit	IEDM Lower Limit	IEDM Mean	IEDM Upper Limit
1	1,89	2,06	2,22	1,88	2,04	2,21	2,06	2,23	2,41
2	2,10	2,30	2,49	2,08	2,28	2,48	2,45	2,67	2,89
3	2,61	2,79	2,96	2,66	2,83	3,01	3,05	3,23	3,42
4	1,04	1,17	1,31	1,05	1,19	1,33	3,15	3,58	4,00
5	2,86	3,04	3,23	2,86	3,04	3,23	3,45	3,65	3,85
6	3,43	3,70	3,96	3,46	3,73	3,99	4,17	4,46	4,75
7	3,88	4,08	4,29	3,88	4,08	4,28	4,34	4,55	4,76
8	15,21	15,92	16,63	15,22	15,94	16,65	16,36	17,08	17,81
9	18,78	19,75	20,71	18,92	19,88	20,85	19,66	20,64	21,62
10	16,61	17,54	18,47	16,56	17,47	18,38	21,71	22,90	24,09
11	18,71	19,93	21,15	18,92	20,16	21,39	24,10	25,41	26,72
12	26,83	29,09	31,34	26,91	29,18	31,45	29,98	32,34	34,70
13	26,36	28,77	31,18	26,20	28,59	30,98	30,74	33,28	35,81
14	26,65	28,48	30,32	26,83	28,66	30,49	35,49	37,26	39,03
15	54,80	57,34	59,88	54,78	57,37	59,96	62,41	65,14	67,86
16	98,13	100,14	102,15	98,04	100,03	102,02	105,86	107,84	109,81

Table 11. Confidence Intervals for linear computable variables

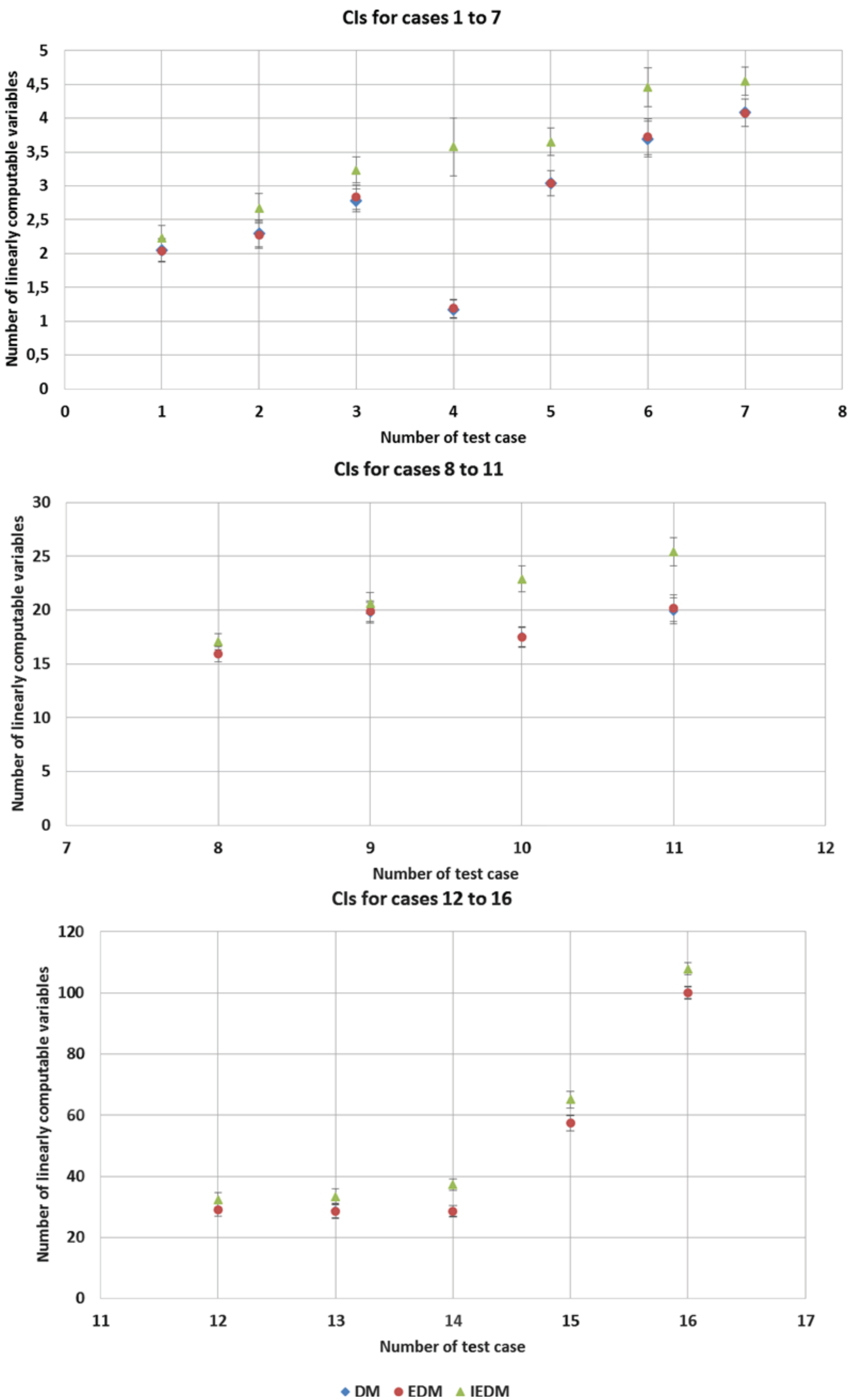


Fig. 13. Charts of Confidence Intervals from Table 11

According to the results reported on Table 11, the new algorithm (IEDM) exhibited a clear improvement with respect to the formation of linear blocks, from the given incidence matrices structure and the Non-Linearity Degrees of each problem. It can be stated that if there are linear blocks, it is more likely to find them through IEDM than by means of the prior methods (DM and EDM). Therefore, this improvement is statistically significant enough.

As can be seen in the results of case studies, all the cases show favourable results to the IEDM. Even better, not only the mean values for linear computable variables are higher for the IEDM, but also the CIs do not overlap for the majority of problem instances. This can be noted in Fig. 13, where the means and error bars are drawn. Only few cases (Feed-15, Feed-9, HeatExchange, CamShape, Robot and Rocket) did not yield statistical significance to conclude about the IEDM performance, but in all of them there is an enhancement in the mean value favouring IEDM. In particular, the problems called Distillation, Methanol, Glider, MTBE-Col, HDA and Ammonia show that significant improvements have been achieved for IEDM. Their mean and CIs values correspond to a substantial increase in the number of additional linearly computable variables with respect to DM and EDM, considering the non-overlapping of CIs.

4.4. Discussion

The software platform is an interactive computing environment that enables the generation of a comprehensive collection of test cases designed for rapid evaluation of partitioning methods. The platform provides flexibility and specialized capabilities to evaluate particular applications, such as simulation, optimization and instrumentation design.

The partitioning methods shown in this article have their origin on several case-studies coming from the Instrumentation Design of Chemical Process Plants. Given that the mathematical models emerging in such area are similar in various structural aspects to other models arising in PSE, many improvements in efficiency over the handling of several models might be achieved. For this purpose, it can be envisaged that there is an open research line for the exploration of the corresponding structural aspects in those models.

An important related issue is the fine-tuning of the statistical parameters that rule over the case-study generation in the described platform. Many of these parameters can be treated in alternative ways. For example, the probabilistic distribution of the mean of indeterminable block size could be switched from Normal to Uniform, if it would fit right to any models. It can be case-dependent; then, further studies might help on the determination of the best parameters for each case.

In the generation of case-studies in the proposed platform, the NLD determination is done only through several random numbers, associated with the equations and variables in the system. In view of better capturing the shape and features of equation systems, a more accurate way to carry out NLD determination might be the random generation of equation terms. Nevertheless, though practical, the generalisation to any equation system might blur the analysis with cases that rarely come out in PSE.

It is interesting to note that the parameters that govern the mathematical models studied here might be similar to those coming from some models that belong to other scientific disciplines.

For example, the incidence matrix that emerges from the mathematical model of a chemical process plant may have the same density and proportion of linear equations as another incidence matrix generated from a physical model. If such were the case, then the results reported in this article might be applied directly to this distinct model. In other words, the conclusions about the statistical validity of a partitioning method drawn in this section are case-independent.

5. Conclusions

A novel interactive software platform has been proposed and implemented to support design and development for PSE. Based on this article and the expected industrial needs, some opportunities can be identified about further computer-aided applications of this platform. It is focused on methods to evaluate alternative decompositions in order to identify the most convenient strategy. Then, this is a design tool for case assessment that may be exploited to enhance research about topics of special interest, such as simulation and optimization, superstructure construction, the systematic enumeration of configurations and multi-scenario formulation.

This platform generates random instances of incidence matrices and other data structures, which are directly related to equations systems associated with mathematical models. Its main objective is the statistical validation of certain structural partitioning methods. In particular, the DM, EDM and IEDM were evaluated in this paper. The IEDM exhibited notable improvements over the remaining two partitioning methods. These enhancements have a good statistical support, as can be inferred from the CIs. Besides, in search for simpler calculation related to finding smaller simpler subsystems that can be solved separately, it might be beneficial to identify the impact of tuning the weight values that serve to obtain the corresponding NLD. It should be kept in mind that a change in the weight values can lead to more appropriate values of NLD, depending on the addressed problem.

The potential for this tool’s contributions to PSE design remains great since it is possible to establish a narrower connection between the cases generated by this platform and real-world modelling. For design-oriented research, this issue remains underexploited. An optimal fitting of the parameters that rule over data-structure arrangement is desirable in order to achieve incidence matrices more closely related to the selected mathematical models. Moreover, in pursuit of a wider, more accurate validation, several models might be employed, establishing the parameters on the basis of their features. The adopted models might originate from different scientific or industrial areas, thus justifying the inclusion of the partitioning methods into already existing software packages. In this way, the partitioning methods chosen ad hoc might become a useful contribution for various modeling tools.

References

Abu-Khzam FN, Fellows MR, Langston MA, Suters WH. Crown structures for vertex cover kernelization. Theory Comput. Syst. 2007; 41(3): 411-30

Bendib R, Khelassi A. Assessment of the effect of feed plate location on interactions for a binary distillation column. World Academy of Science, Engineering and Technology 2006; 24.

Bike S. Design of an ammonia synthesis plant. CACHE Case Study. Department of Chemical Engineering, Carnegie Mellon University; 1985.

Bomhoff M, Kern W, Still G. On bounded block decomposition problems for under-specified systems of equations. J. Comp. Syst. Sci. 2012; 78(1): 336-47.

Brooke A, Kendrick D, Meeraus A. GAMS: a user's guide. Release 2.25. Massachusetts:Boyd & Fraser Publishing Company; 1992.

Cañete B, Gigola CE, Brignole NB. Synthesis gas processes for methanol production via CH_4 reforming with CO_2, H_2O and O_2 Ind. & Eng. Chem. Res. 2014; 53 (17): 7103–12.

Chlebík M, Chlebíková J. Crown reductions for the Minimum Weighted Vertex Cover problem. Discrete Applied Math. 2008; 156: 292-312

Cucek L, Martín M, Grossmann IE, Kravanja Z. Energy, water and process technologies integration for the simultaneous production of ethanol and food from the entire corn plant. Comp. & Chem. Eng. 2011; 35(8): 1547-57.

Dolan ED, Moré JJ, Munson TS. Benchmarking optimization software with COPS 3.0. Technical Report ANL/MCS-TM-273, Argonne national laboratory, Mathematics and Computer Science division; 2004.

Domancich A, Ardenghi JI, Vazquez GE, Brignole NB. Desempeño de un algoritmo espectral para optimización rigurosa de plantas de procesos industriales. Mec Comput. 2004; 23:3047-57.

Domancich AO, Durante M, Ferraro S, Hoch P, Brignole NB, Ponzoni I. How to improve the model partitioning in a DSS for instrumentation design. Ind. & Eng. Chem. Res. 2009; 48(7): 3513-25.

Drummond LA, Duff IS, Guivarch R, Ruiz D, Zenadi M. Partitioning strategies for the Block Cimmino algorithm. Journal of Engineering Mathematics. 2015; 93 (1): 21-39

Duff IS, Reid JK. An implementation of Tarjan’s algorithm for the block triangularization of a matrix. ACM Trans. Math. Softw. 1978; 4: 137-47.

Duff IS, Reid JK. Some design features of a sparse matrix code. ACM Trans. on Math. Soft. 1979; 5(1): 18-35.

Duff IS. A survey of sparse matrix research. Proc. IEEE 1977; 65: 500-35.

Edgar TF, Himmelblau DM, Lasdon L. Optimization of chemical processes. McGraw-Hill; 1989.

Fleischer LK, Hendrickson B, Pinar A. On identifying Strongly Connected Components in parallel. Parallel and Distributed Processing, IPDPS Workshops, Lecture Notes in Computer Science. 2000; 1800: 505–11

Heath MT. Scientific Computing. An Introductory Survey. 2nd Ed., Mc Graw Hill; 2002.

Hopcroft J, Karp R. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J. Comp. 1973; 2(4): 225-31.

Jacobs R, Krishna R. Multiple solutions in reactive distillation for methyl tert-butyl ether synthesis. Ind. & Eng. Chem. Res. 1993; 32(8), 1706-1709.

Kheawhom S, Hirao M. Decision support tools for process design and selection. Comp. & Chem. Eng., 2002; 26(4), 747-755.

Lowe G. Concurrent depth-first search algorithms based on Tarjan's Algorithm, TACAS 2014, Int J Softw Tools Technol Transfer, 2015; 1-19

Nilsson H, Peterson J, Hudak P. Functional hybrid modeling from an object-oriented perspective. In: EOOLT., 2007.

Ponzoni I, Sánchez MC, Brignole NB. Direct Method for structural observability analysis. Ind. & Eng. Chem. Res. 2004; 43(2): 577-88.

Qiu QF, Rangaiah GP, Krishnaswamy PR. Application of a plant-wide control design to the HDA process. Comp. & Chem. Eng. 2003; 27(1), 73-94.

Sargent RWH, Westerbeg AW. "Speed-up" in chemical engineering design. Trans. Inst. Chem. Eng. 1964; 42: 190- 97

Tarjan R. Depth-First Search and linear graph algorithms. SIAM J. Comp. 1972; (1): 146-60.

Tjoa IB, Biegler LT. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. Ind. & Eng. Chem. Res. 1991; 30(2): 376-85.

Xamena E, Maguitman AG, Brignole NB. Optimized resolution of systems of equations. In: XVIII Congreso Argentino de Ciencias de la Computación, 2012.

Table1 COLOR

Case Study	Amount of	DM	EDM	Δ
Distillation Column	observable variables	63	63	0
	unobservable variables	22	22	0
	1 × 1 subsystems	48	58	+21%
	linear subsystems	47	57	+21%
	nonlinear subsystems	4	2	-50%
Ammonia Synthesis Plant	observable variables	216	216	0
	unobservable variables	297	297	0
	1 × 1 subsystems	144	164	+14%
	linear subsystems	143	164	+15%
	nonlinear subsystems	8	4	-50%
Ethane Plant	observable variables	929	929	0
	unobservable variables	673	673	0
	1 × 1 subsystems	767	777	+1.3%
	linear subsystems	758	776	+2.3%
	nonlinear subsystems	26	20	-23%

Table2 COLOR

	Variables	Equations		Variables	Equations
Size 1	10	8	Size 2	9	4
	4	7		6	5
	9	12		2	3
	9	5		4	11
	8	12		4	4
	7	6		1	11
	3	8			
	4	12			
	5	12			
	7	9			
	10	12			
	9	2			
	9	6			
	3	7			

Table3 COLOR

	x_2	x_{10}	x_7	x_6	x_8	x_1	x_5	x_3	x_4	x_9	Block Type	Group
e_7	1	0	1	0	0	0	0	0	0	0	LINEAR	SR1
e_4	1	1	1	0	0	0	0	0	0	0		
e_1	1	1	1	0	0	0	0	0	0	0		
e_{10}	1	1	1	1	0	0	0	0	0	0	LINEAR	
e_3	1	0	0	0	1	0	0	0	0	0	LINEAR	SR2
e_{11}	1	1	1	1	1	1	0	0	0	0	NONLINEAR	
e_6	1	0	0	0	0	1	1	0	0	0	NONLINEAR	
e_2	1	0	0	0	0	0	0	0	0	0		VR
e_5	0	1	0	1	0	0	0	0	0	0		
e_8	0	0	1	1	0	0	0	0	0	0		
e_9	1	1	1	1	0	0	0	0	0	0		
e_{12}	1	1	1	1	0	0	1	1	1	1		HR
Group	SC1				SC2			HC				

Table 4 COLOR

	DM	EDM	IEDM
Linear Blocks	1	2	3
Variables computable by linear blocks	1	3	5
Nonlinear Blocks	3	4	2
Variables computable by nonlinear blocks	6	4	2
Total number of blocks	4	6	5
Total number of observable variables	7	7	7

Table5 COLOR

Parameter	Distillation Column	Ammonia Plant
Equations	102	557
Variables	85	513
Number of Cells	8670	285741
Number of Ones	265	1991
Density	3.06%	0.70%
Number of forbidden blocks	29	104
Maximum Forbidden Block Size	10	21
Percentage of linear equations	60%	54%

Table 6 COLOR

Nº	Name	Eq. #	Total # of Variables	Density	Proportion of linear Equations
1	Feed-15	138	137	3,32%	24,64%
2	Feed-9	84	83	5,29%	26,19%
3	Chloride	32	33	10,23%	34,38%
4	Distillation	102	85	3,06%	60,78%
5	Gasnet	49	58	5,03%	8,16%
6	Catmix	201	227	1,98%	0,50%
7	PropGlycol	28	29	10,71%	42,86%
8	Heat-Exchangers	58	59	2,98%	89,66%
9	Camshape	100	135	1,51%	100,00%
10	Methanol	367	312	1,29%	44,41%
11	Glider	610	612	0,60%	8,69%
12	Robot	403	484	0,74%	49,88%
13	Rocket	503	525	0,74%	44,14%
14	MTBE-Column	1923	1817	0,22%	21,53%
15	HDA	639	704	0,44%	75,27%
16	Ammonia	557	513	0,70%	54,04%

Table 7 COLOR

Partitioning Method	Mean	Std Dev	CI 95%
DM	0,83	1,450	0,546 - 1,114
EDM	0,78	1,411	0,503 - 1,057
IEDM	2,47	5,500	1,392 - 3,548

Table 8 COLOR

Partitioning Method	**Mean**	**Std Dev**	**CI 95%**
DM	1,17	2,148	1,041 - 1,307
EDM	1,19	2,213	1,055 - 1,329
IEDM	3,57	6,885	3,148 - 4,002

Table 9 COLOR

Partitioning Method	Mean	Std Dev	CI 95%
DM	103,54	34,137	96,84 - 110,23
EDM	103,80	35,126	96,92 - 110,68
IEDM	108,10	33,388	101,56 - 114,64

Table 10 COLOR

Partitioning Method	Mean	Std Dev	CI 95%
DM	100,14	32,428	98,13 - 102,15
EDM	100,03	32,141	98,04 - 102,02
IEDM	107,84	31,892	105,86 - 109,81

Table 11 COLOR

Nº	DM Lower Limit	DM Mean	DM Upper Limit	EDM Lower Limit	EDM Mean	EDM Upper Limit	IEDM Lower Limit	IEDM Mean	IEDM Upper Limit
1	1,89	2,06	2,22	1,88	2,04	2,21	2,06	2,23	2,41
2	2,10	2,30	2,49	2,08	2,28	2,48	2,45	2,67	2,89
3	2,61	2,79	2,96	2,66	2,83	3,01	3,05	3,23	3,42
4	1,04	1,17	1,31	1,05	1,19	1,33	3,15	3,58	4,00
5	2,86	3,04	3,23	2,86	3,04	3,23	3,45	3,65	3,85
6	3,43	3,70	3,96	3,46	3,73	3,99	4,17	4,46	4,75
7	3,88	4,08	4,29	3,88	4,08	4,28	4,34	4,55	4,76
8	15,21	15,92	16,63	15,22	15,94	16,65	16,36	17,08	17,81
9	18,78	19,75	20,71	18,92	19,88	20,85	19,66	20,64	21,62
10	16,61	17,54	18,47	16,56	17,47	18,38	21,71	22,90	24,09
11	18,71	19,93	21,15	18,92	20,16	21,39	24,10	25,41	26,72
12	26,83	29,09	31,34	26,91	29,18	31,45	29,98	32,34	34,70
13	26,36	28,77	31,18	26,20	28,59	30,98	30,74	33,28	35,81
14	26,65	28,48	30,32	26,83	28,66	30,49	35,49	37,26	39,03
15	54,80	57,34	59,88	54,78	57,37	59,96	62,41	65,14	67,86
16	98,13	100,14	102,15	98,04	100,03	102,02	105,86	107,84	109,81

Table 1 BW

Case Study	Amount of	DM	EDM	Δ
Distillation Column	observable variables	63	63	0
	unobservable variables	22	22	0
	1 × 1 subsystems	48	58	+21%
	linear subsystems	47	57	+21%
	nonlinear subsystems	4	2	-50%
Ammonia Synthesis Plant	observable variables	216	216	0
	unobservable variables	297	297	0
	1 × 1 subsystems	144	164	+14%
	linear subsystems	143	164	+15%
	nonlinear subsystems	8	4	-50%
Ethane Plant	observable variables	929	929	0
	unobservable variables	673	673	0
	1 × 1 subsystems	767	777	+1.3%
	linear subsystems	758	776	+2.3%
	nonlinear subsystems	26	20	-23%

Table 2 BW

	Variables	Equations		Variables	Equations
Size 1	10	8	Size 2	9	4
	4	7		6	5
	9	12		2	3
	9	5		4	11
	8	12		4	4
	7	6		1	11
	3	8			
	4	12			
	5	12			
	7	9			
	10	12			
	9	2			
	9	6			
	3	7			

Table 3 BW

	x_2	x_{10}	x_7	x_6	x_8	x_1	x_5	x_3	x_4	x_9	Block Type	Group
e_7	1	0	1	0	0	0	0	0	0	0	LINEAR	SR1
e_4	1	1	1	0	0	0	0	0	0	0		
e_1	1	1	1	0	0	0	0	0	0	0		
e_{10}	1	1	1	1	0	0	0	0	0	0	LINEAR	
e_3	1	0	0	0	1	0	0	0	0	0	LINEAR	SR2
e_{11}	1	1	1	1	1	1	0	0	0	0	NONLINEAR	
e_6	1	0	0	0	0	1	1	0	0	0	NONLINEAR	
e_2	1	0	0	0	0	0	0	0	0	0		VR
e_5	0	1	0	1	0	0	0	0	0	0		
e_8	0	0	1	1	0	0	0	0	0	0		
e_9	1	1	1	1	0	0	0	0	0	0		
e_{12}	1	1	1	1			1	1	1	1		HR
Group	SC1				SC2			HC				

Table 4 BW

	DM	EDM	IEDM
Linear Blocks	1	2	3
Variables computable by linear blocks	1	3	5
Nonlinear Blocks	3	4	2
Variables computable by nonlinear blocks	6	4	2
Total number of blocks	4	6	5
Total number of observable variables	7	7	7

Table 5 BW

Parameter	Distillation Column	Ammonia Plant
Equations	102	557
Variables	85	513
Number of Cells	8670	285741
Number of Ones	265	1991
Density	3.06%	0.70%
Number of forbidden blocks	29	104
Maximum Forbidden Block Size	10	21
Percentage of linear equations	60%	54%

Table 6 BW

Nº	Name	Eq. #	Total # of Variables	Density	Proportion of linear Equations
1	Feed-15	138	137	3,32%	24,64%
2	Feed-9	84	83	5,29%	26,19%
3	Chloride	32	33	10,23%	34,38%
4	Distillation	102	85	3,06%	60,78%
5	Gasnet	49	58	5,03%	8,16%
6	Catmix	201	227	1,98%	0,50%
7	PropGlycol	28	29	10,71%	42,86%
8	Heat-Exchangers	58	59	2,98%	89,66%
9	Camshape	100	135	1,51%	100,00%
10	Methanol	367	312	1,29%	44,41%
11	Glider	610	612	0,60%	8,69%
12	Robot	403	484	0,74%	49,88%
13	Rocket	503	525	0,74%	44,14%
14	MTBE-Column	1923	1817	0,22%	21,53%
15	HDA	639	704	0,44%	75,27%
16	Ammonia	557	513	0,70%	54,04%

Table 7 BW

Partitioning Method	Mean	Std Dev	CI 95%
DM	0,83	1,450	0,546 - 1,114
EDM	0,78	1,411	0,503 - 1,057
IEDM	2,47	5,500	1,392 - 3,548

Table 8 BW

Partitioning Method	Mean	Std Dev	CI 95%
DM	1,17	2,148	1,041 - 1,307
EDM	1,19	2,213	1,055 - 1,329
IEDM	3,57	6,885	3,148 - 4,002

Table 9 BW

Partitioning Method	Mean	Std Dev	CI 95%
DM	103,54	34,137	96,84 - 110,23
EDM	103,80	35,126	96,92 - 110,68
IEDM	108,10	33,388	101,56 - 114,64

Table 10 BW

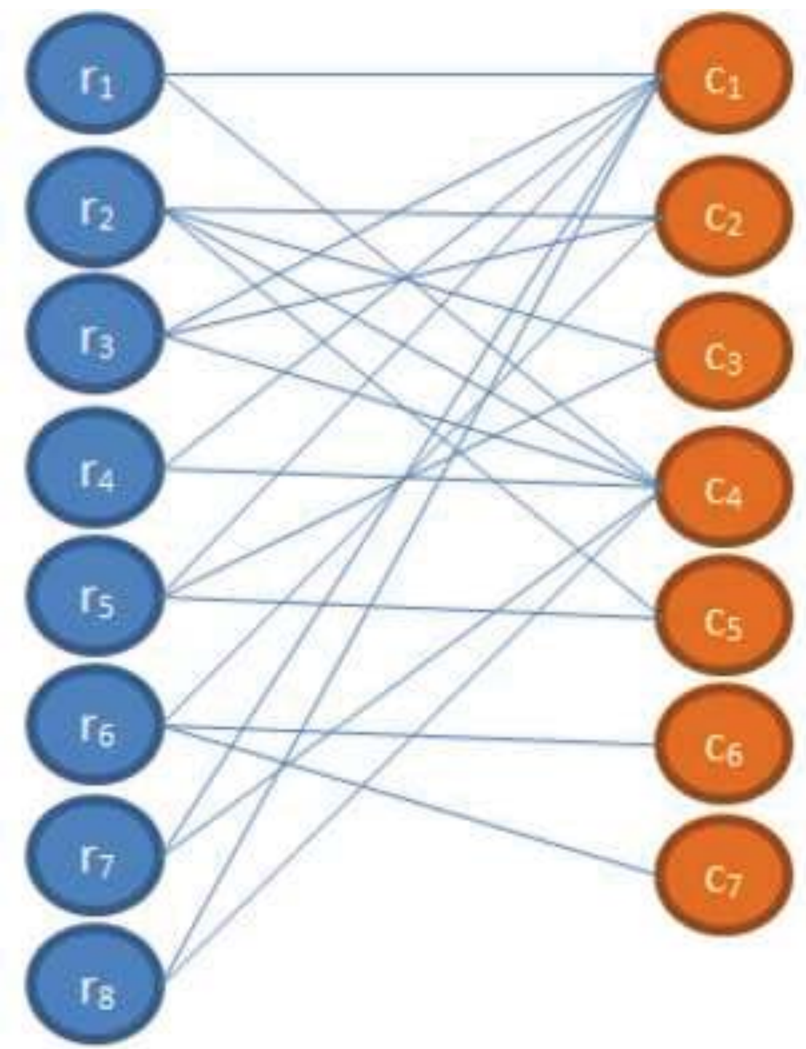
Partitioning Method	Mean	Std Dev	CI 95%
DM	100,14	32,428	98,13 - 102,15
EDM	100,03	32,141	98,04 - 102,02
IEDM	107,84	31,892	105,86 - 109,81

Table 11 BW

Nº	DM Lower Limit	DM Mean	DM Upper Limit	EDM Lower Limit	EDM Mean	EDM Upper Limit	IEDM Lower Limit	IEDM Mean	IEDM Upper Limit
1	1,89	2,06	2,22	1,88	2,04	2,21	2,06	2,23	2,41
2	2,10	2,30	2,49	2,08	2,28	2,48	2,45	2,67	2,89
3	2,61	2,79	2,96	2,66	2,83	3,01	3,05	3,23	3,42
4	1,04	1,17	1,31	1,05	1,19	1,33	3,15	3,58	4,00
5	2,86	3,04	3,23	2,86	3,04	3,23	3,45	3,65	3,85
6	3,43	3,70	3,96	3,46	3,73	3,99	4,17	4,46	4,75
7	3,88	4,08	4,29	3,88	4,08	4,28	4,34	4,55	4,76
8	15,21	15,92	16,63	15,22	15,94	16,65	16,36	17,08	17,81
9	18,78	19,75	20,71	18,92	19,88	20,85	19,66	20,64	21,62
10	16,61	17,54	18,47	16,56	17,47	18,38	21,71	22,90	24,09
11	18,71	19,93	21,15	18,92	20,16	21,39	24,10	25,41	26,72
12	26,83	29,09	31,34	26,91	29,18	31,45	29,98	32,34	34,70
13	26,36	28,77	31,18	26,20	28,59	30,98	30,74	33,28	35,81
14	26,65	28,48	30,32	26,83	28,66	30,49	35,49	37,26	39,03
15	54,80	57,34	59,88	54,78	57,37	59,96	62,41	65,14	67,86
16	98,13	100,14	102,15	98,04	100,03	102,02	105,86	107,84	109,81

Figure 1 COLOR

Click here to download high resolution image



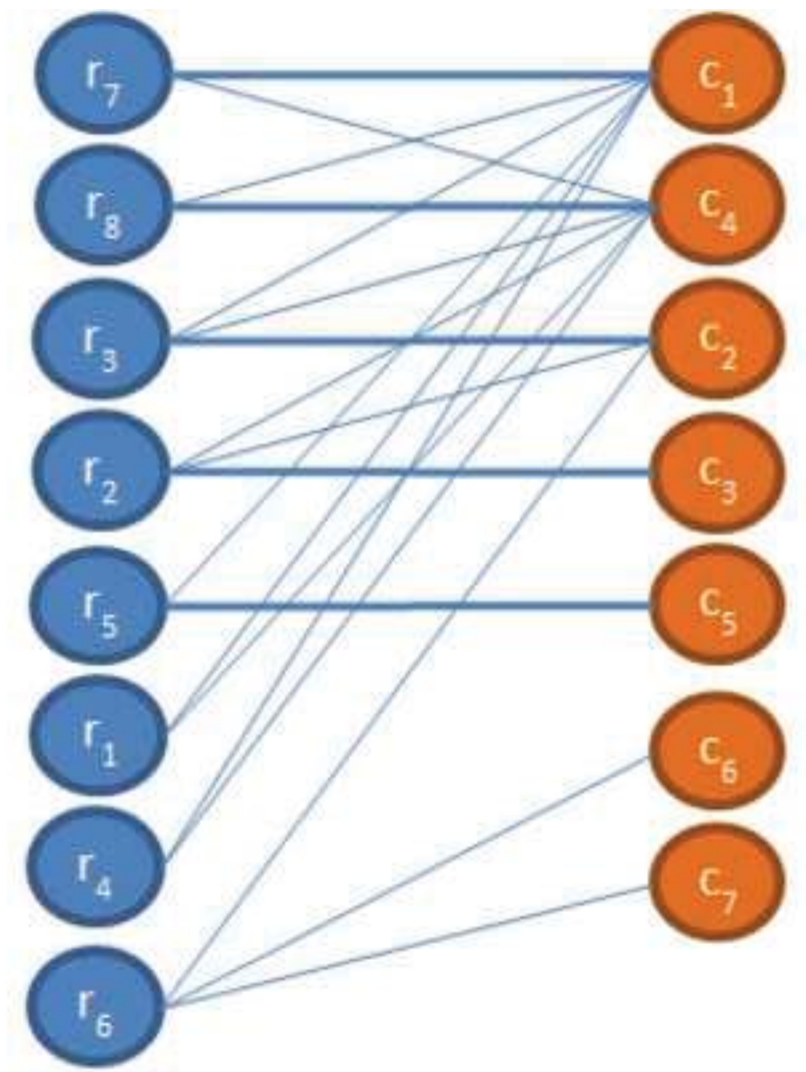


Figure 2 COLOR
Click here to download high resolution image

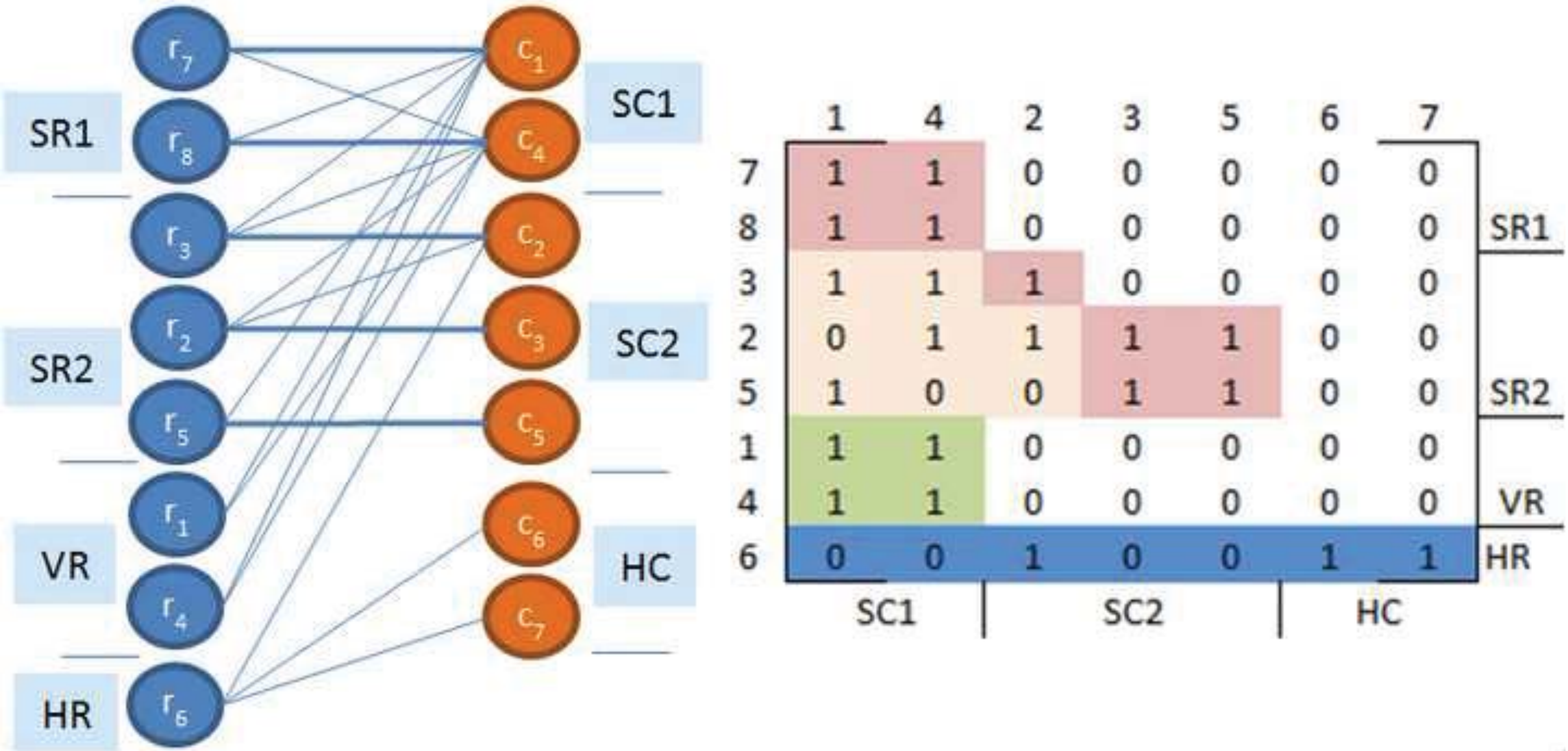
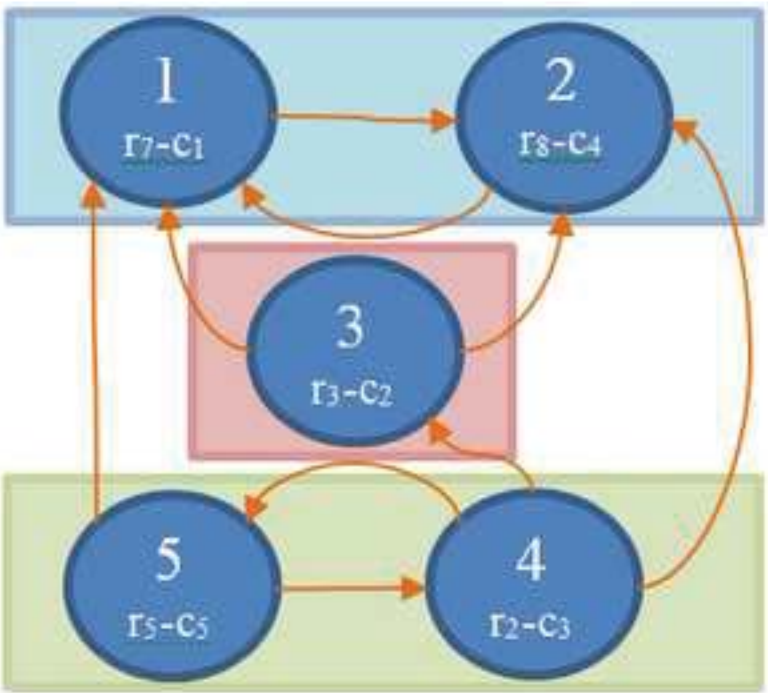


Figure 3 COLOR

Click here to download high resolution image



	1 (1)	4 (2)	2 (3)	3 (4)	5 (5)	
7 (1)	1	1	0	0	0	SCC 1
8 (2)	1	1	0	0	0	
3 (3)	1	1	1	0	0	SCC 2
2 (4)	0	1	1	1	1	SCC 3
5 (5)	1	0	0	1	1	

Figure 4 COLOR

Click here to download high resolution image

$$S_1: \begin{cases} e_1: x_2 + 3x_7 - 2x_{10} - 10 = 0 \\ e_2: \log_{10}(x_2^3 - 17) + x_2^2 - 10 = 0 \\ e_3: -3x_2 - 3x_8 + 15 = 0 \\ e_4: 3x_2 - x_7 - x_{10} = 0 \\ e_5: x_6 + x_{10}^2 + e^{x_{10}-4} - 14 = 0 \\ e_6: 7x_1 - x_5 + \frac{x_2^3}{3} + x_2^2 - 23 = 0 \\ e_7: 2x_2 - 3x_7 + 9 = 0 \\ e_8: x_6 + x_7 - 2 = 0 \\ e_9: 3x_2 - x_6 + x_7 + 2x_{10} - 25 = 0 \\ e_{10}: 2x_2 + x_6 - 2x_7 - x_{10} + 11 = 0 \\ e_{11}: -8x_1 + 15x_6 - 20x_7 - x_{10} + x_2^4 * x_8 + x_2^2 - 14 = 0 \\ e_{12}: x_2^3 - e^{(x_2-3)} - x_3 - 4x_4 - x_5 + x_6 - x_7 + x_9 - x_{10} + 4 = 0 \end{cases}$$

Figure 5 COLOR
Click here to download high resolution image

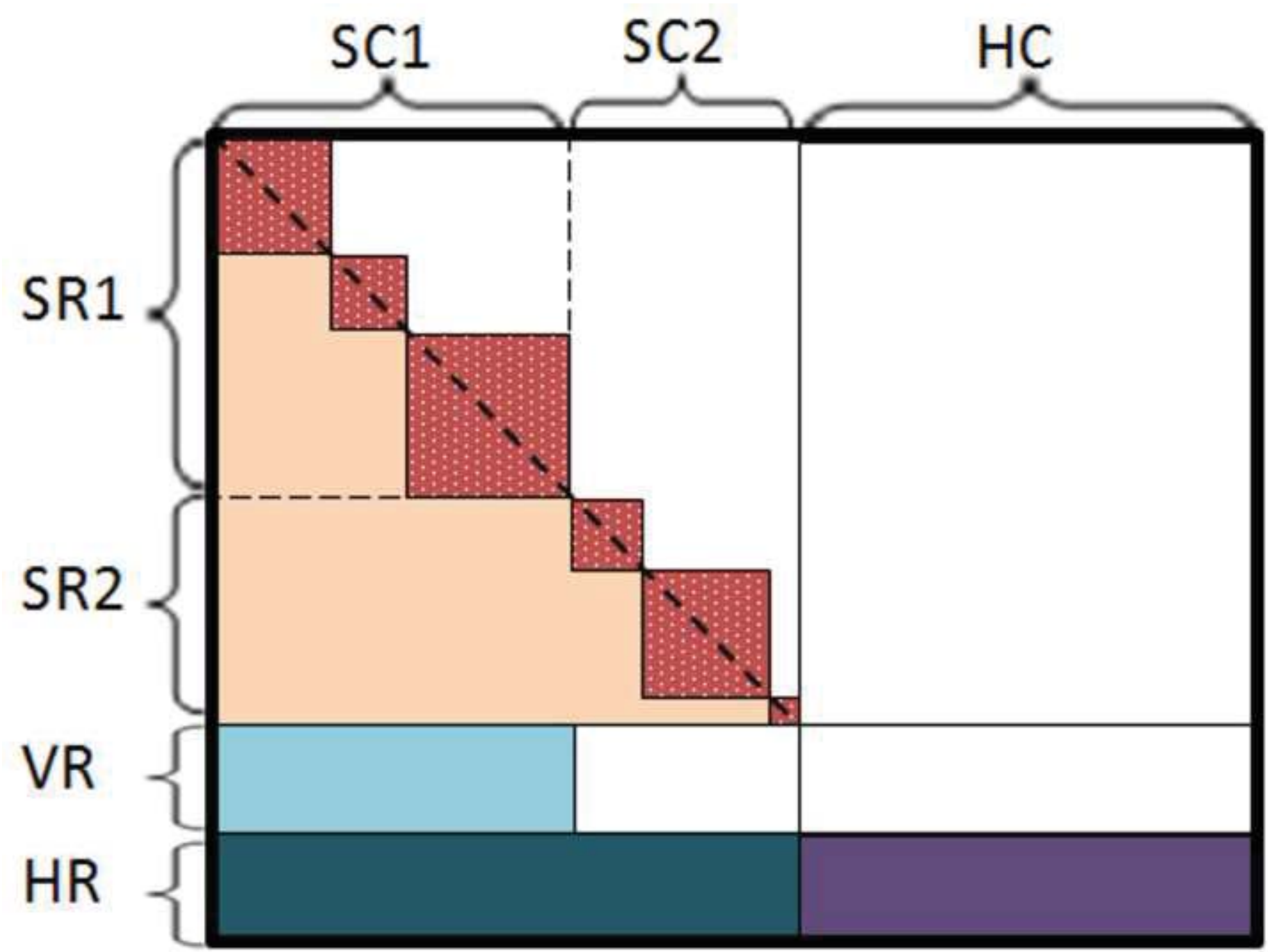


Figure 6 COLOR
Click here to download high resolution image

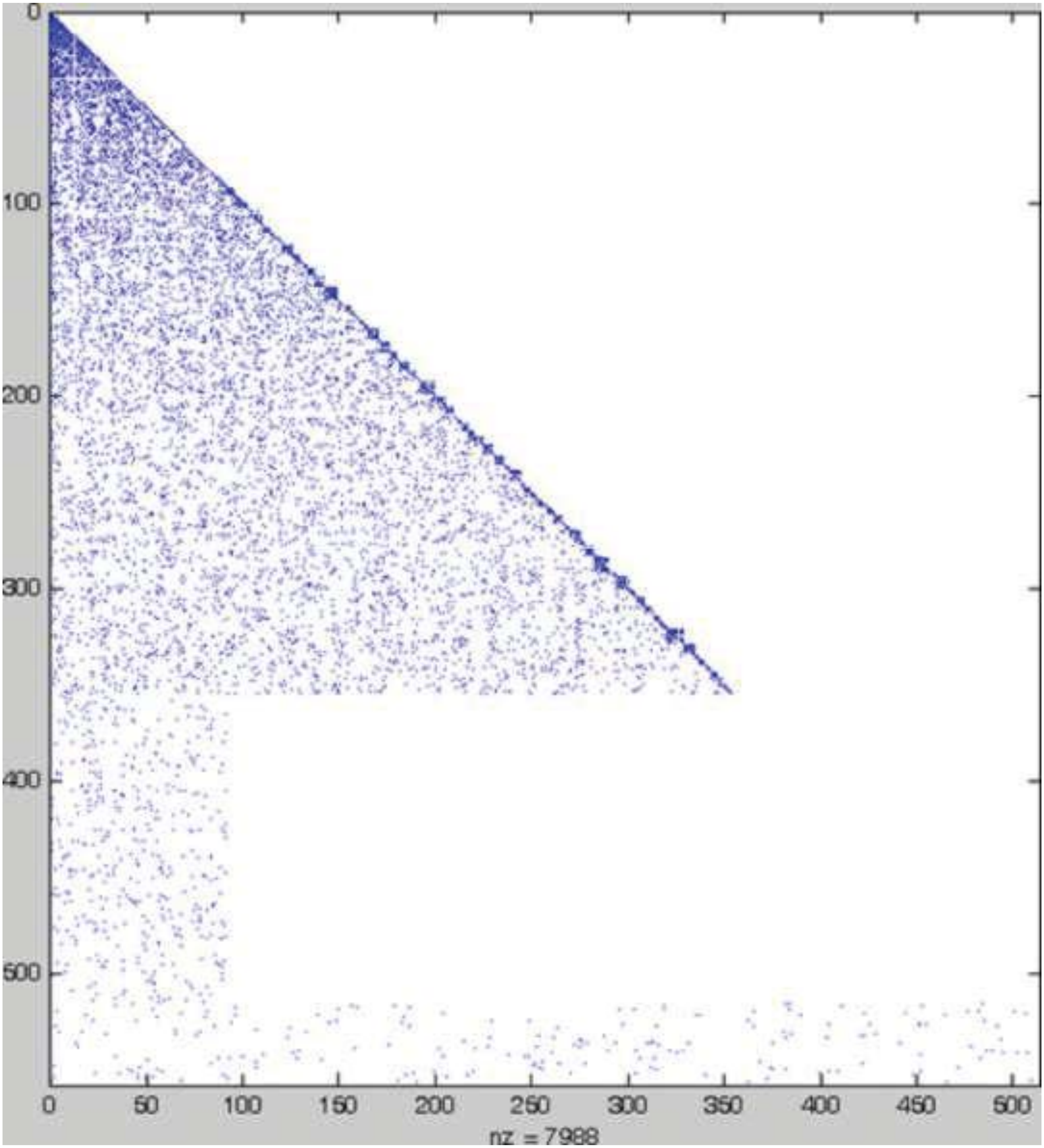


Figure 7 COLOR

Click here to download high resolution image

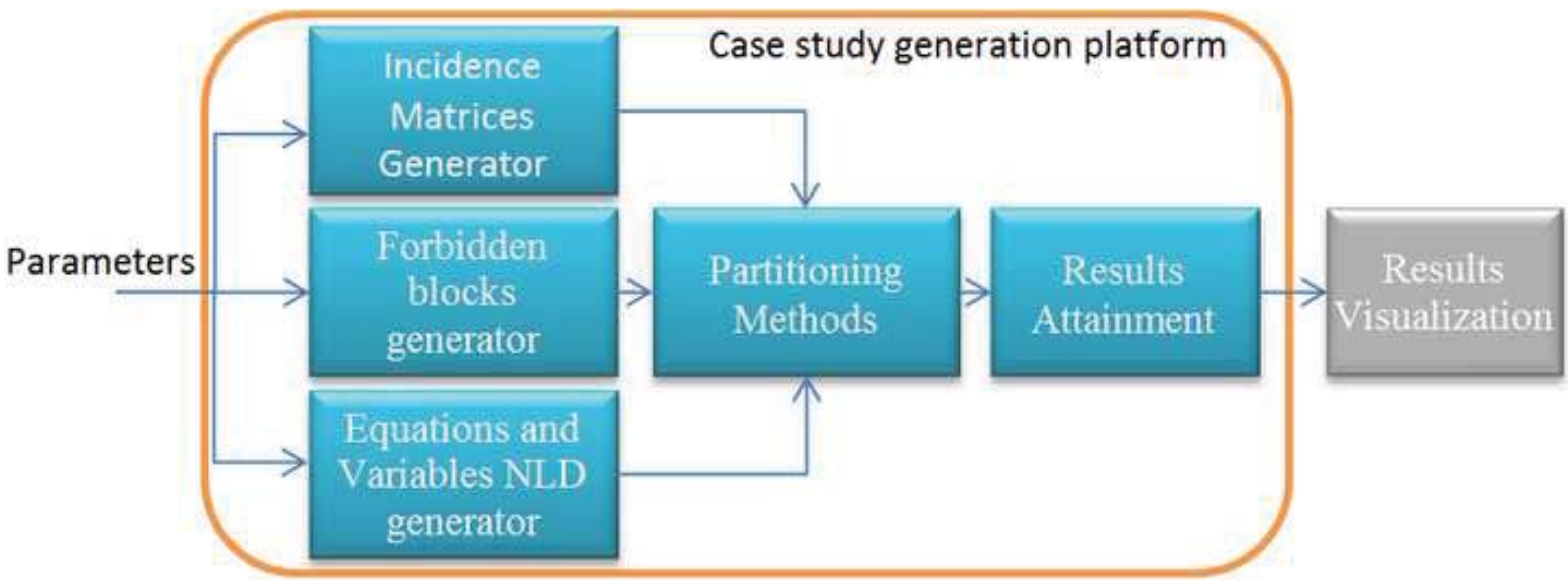
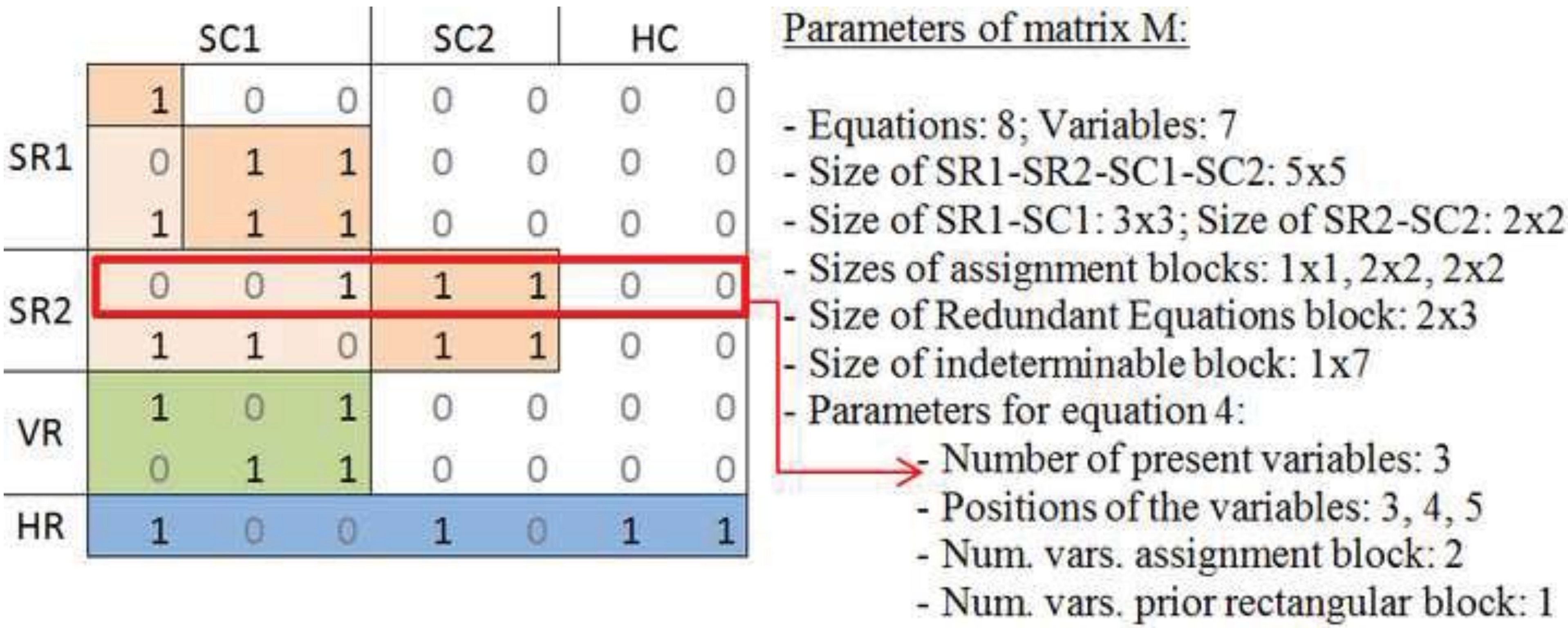


Figure 8 COLOR
Click here to download high resolution image



1. *Input: Parameters for incidence matrices building, Q: Number of cases to build*
2. *Repeat Q times:*

2a. *Generate randomly a matrix M in LBTF, based on the input parameters*

2b. *Randomly reorder M rows and columns*

2c. *Generate a set R of constraints for the assignment blocks formation, based on input parameters*

2d. *Define* ***random*** *NLD for equations and variables*

2e. *NLD=****0*** *for a random group of equations and variables, according to the input parameters*

2f. *Run DM, EDM and IEDM for the incidence matrix and the data structures attained*

2g. *Save partitioning methods results*

1. *Input: Params (Statistical parameters for matrix building)*
2. *sizeBlockSR1SR2=RandomInteger(Params->SR1SR2)*
3. *sizeBlockSR1= RandomInteger (Params->SR1)*
4. *sizeBlockSR2= sizeBlockSR1SR2- sizeBlockSR1*
5. *Diagonal(**BlockSR1SR2**)=1*

Assignment Blocks Definition

6. ***Repeat***

6a. *Randomly generate a size for an assignment block*

7. ***Until SR1-SC1 is complete***
8. ***Repeat***

8a. *Randomly generate a size for an assignment block*

9. ***Until SR2-SC2 is complete***
10. ***For each assignment block in SR1-SC1 and SR2-SC2***

10a. ***For each equation (row)***

10a.i. *Set number of variables of the equation and their positions, considering block bounds*

Redundant Equations Block and Indeterminable Block

11. *IndetBlockColumns=Params. Variables-sizeBlockSR1SR2*
12. *IndetBlockRows= RandomInteger (IndetBlockColumns -1)*
13. *RedunBlockColumns=sizeBlockSR1;*
14. *RedunBlockRows=Params.Equations-sizeBlockSR1SR2- IndetBlockRows;*
15. ***For each equation (row) of Redundant Block***

15a. *Set number of variables of the equation and their positions, considering block bounds*

16. ***For each equation (row) of the Indeterminable Block***

16a. *Set number of variables of the equation and their positions, considering that each equation must have at least 2 indeterminable variables*

Figure 11 COLOR
Click here to download high resolution image

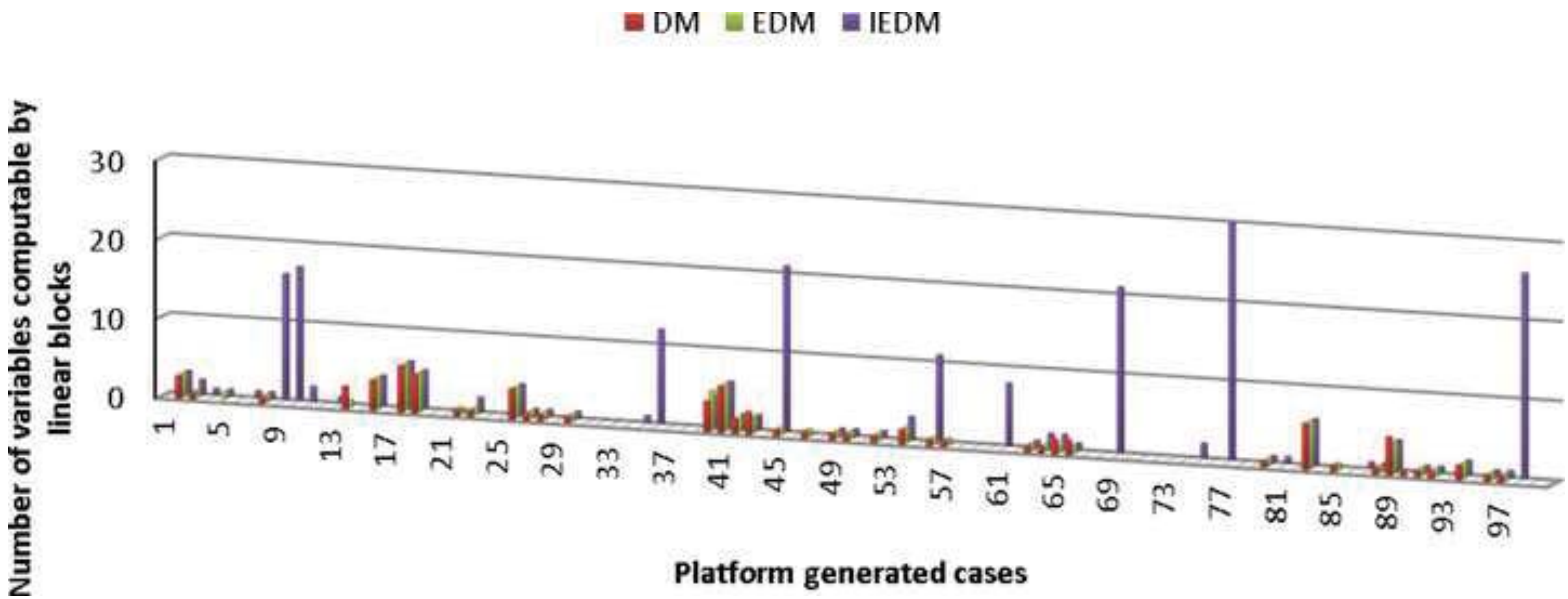


Figure 12 COLOR

Click here to download high resolution image

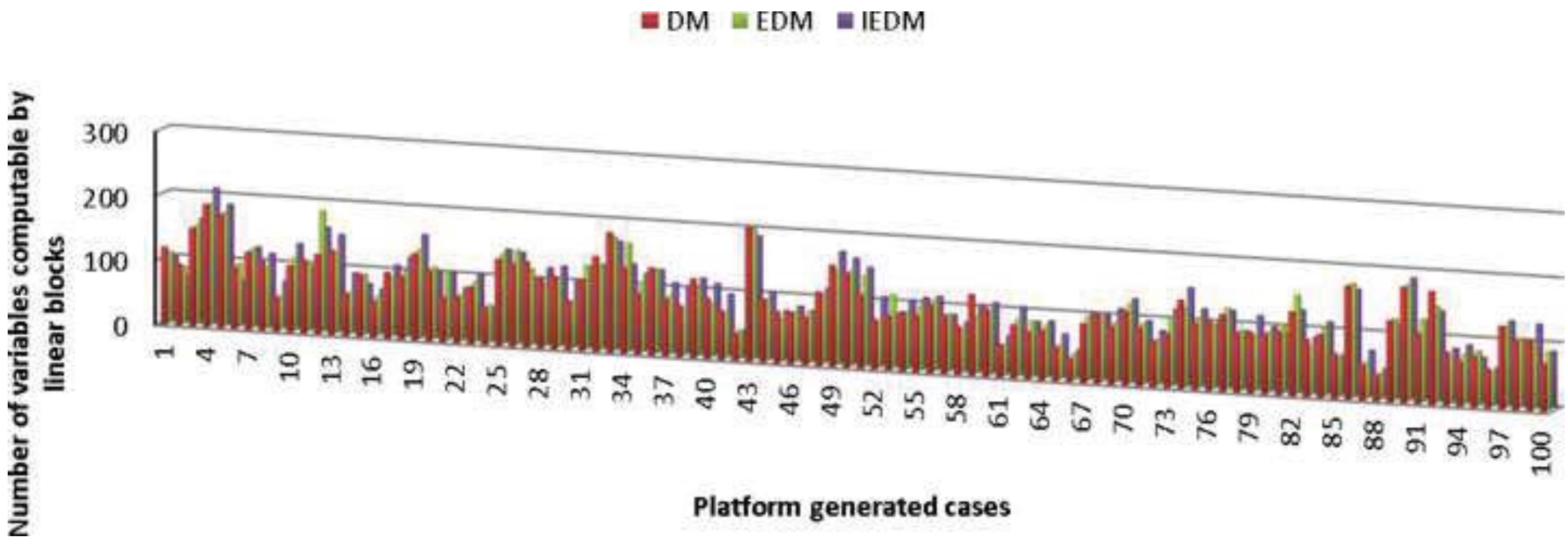


Figure 13 COLOR
Click here to download high resolution image

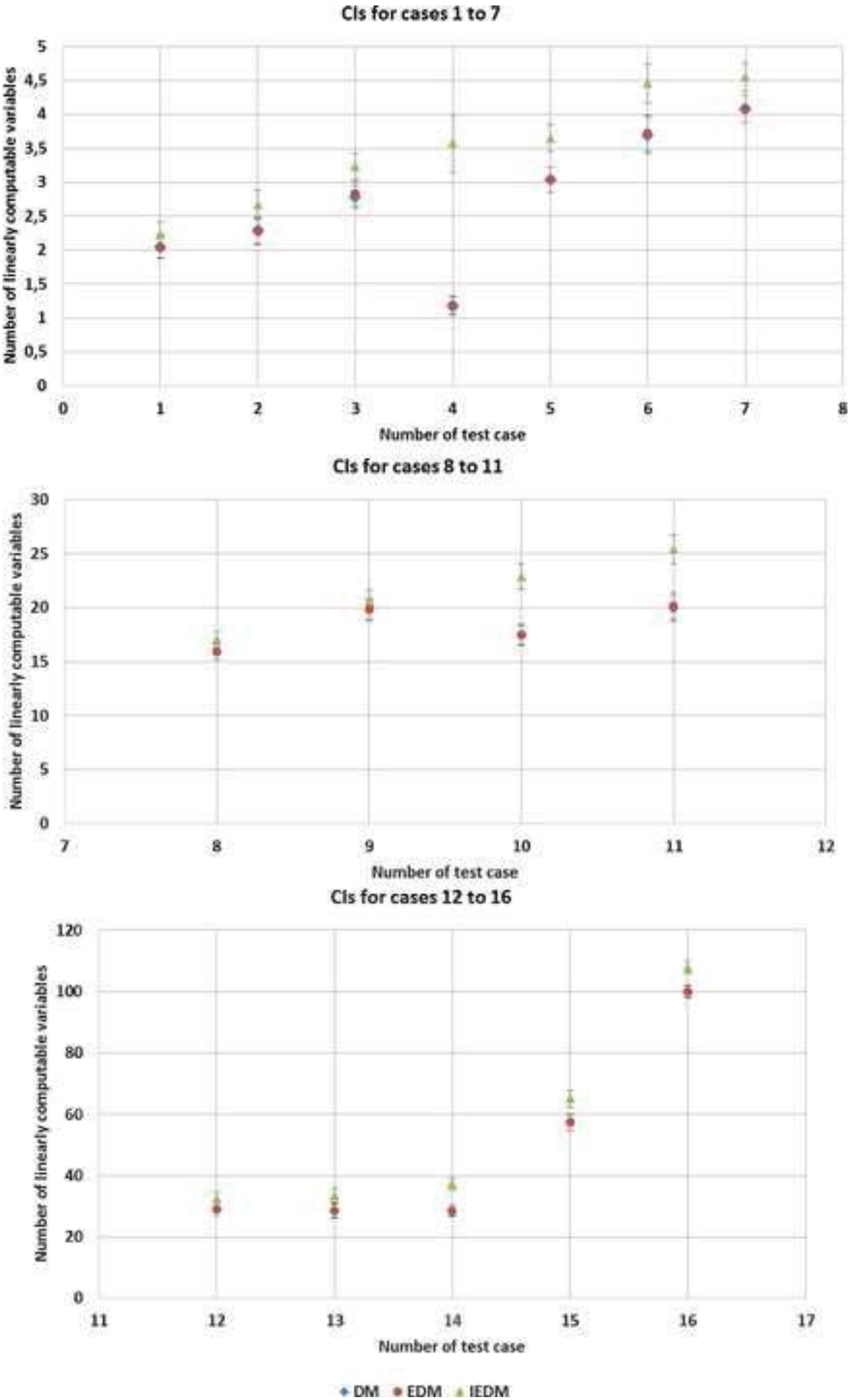


Figure 1 BW

Click here to download high resolution image

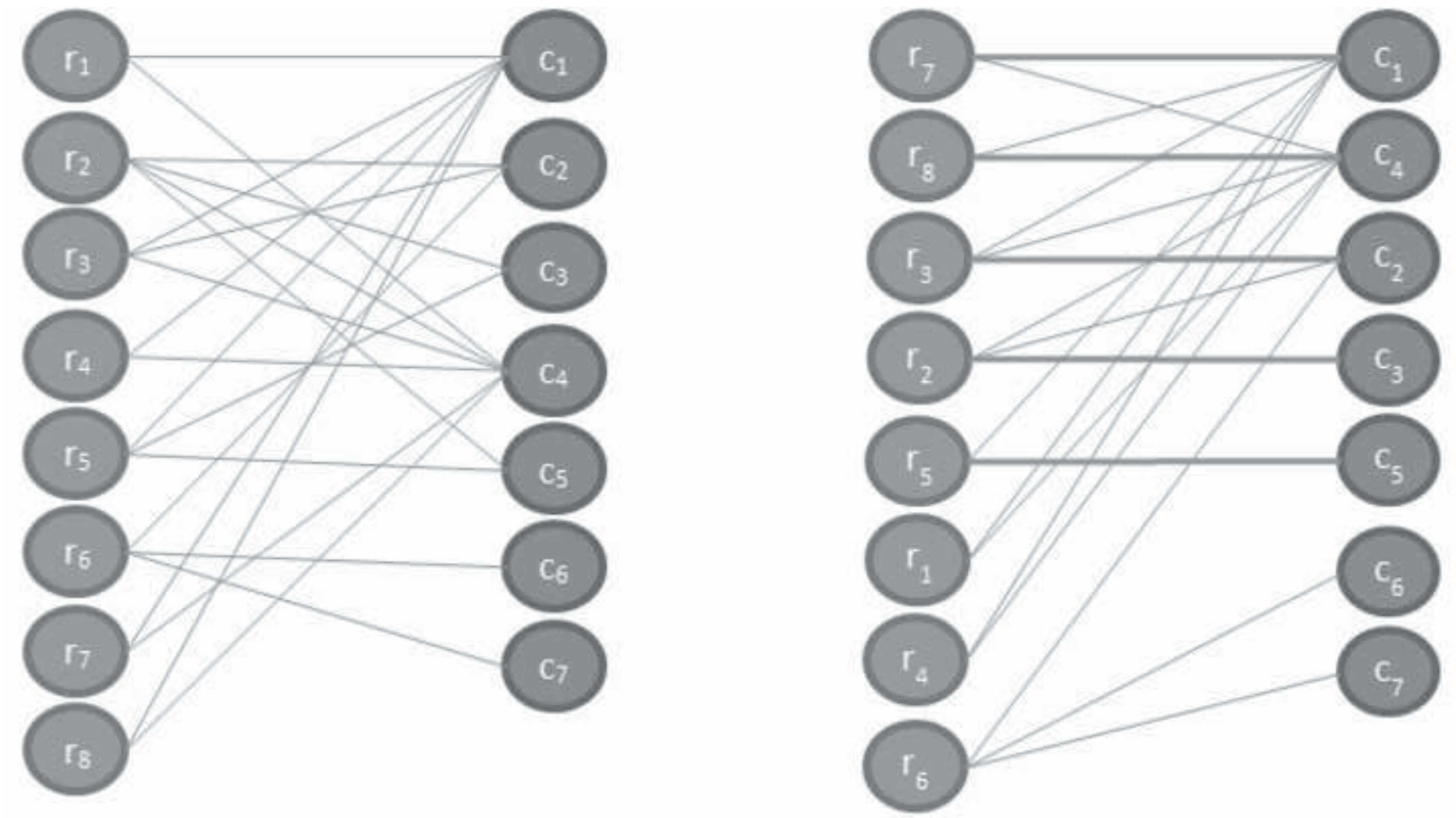
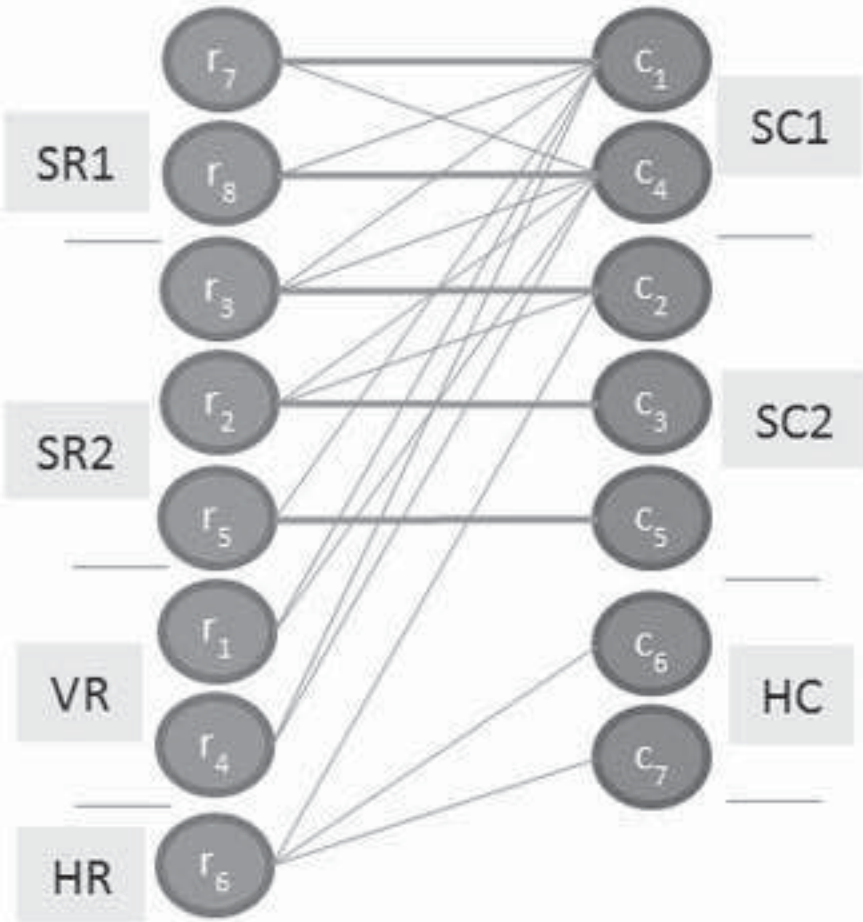


Figure 2 BW

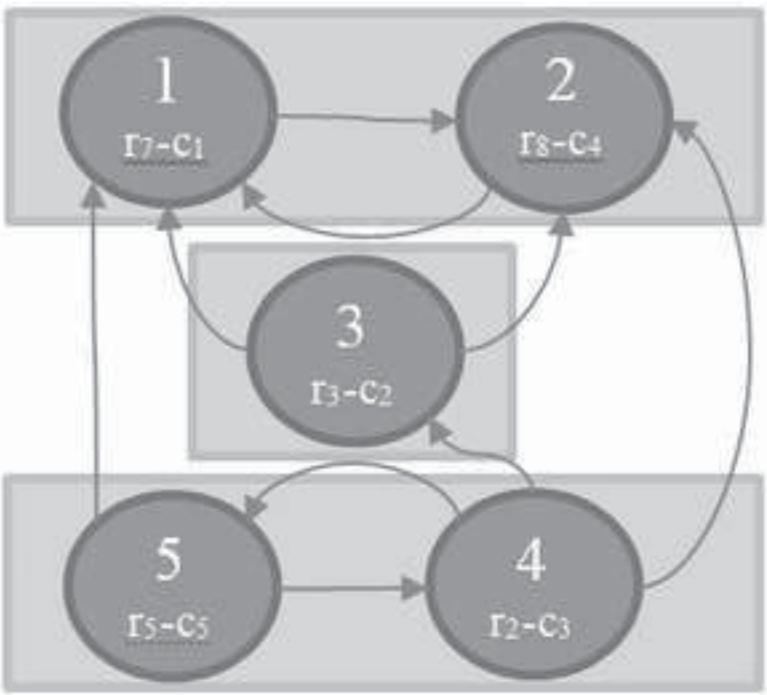
Click here to download high resolution image



	1	4	2	3	5	6	7	
7	1	1	0	0	0	0	0	
8	1	1	0	0	0	0	0	SR1
3	1	1	1	0	0	0	0	
2	0	1	1	1	1	0	0	
5	1	0	0	1	1	0	0	SR2
1	1	1	0	0	0	0	0	
4	1	1	0	0	0	0	0	VR
6	0	0	1	0	0	1	1	HR
	SC1		SC2			HC		

Figure 3 BW

Click here to download high resolution image



	1 (1)	4 (2)	2 (3)	3 (4)	5 (5)	
7 (1)	1	1	0	0	0	SCC 1
8 (2)	1	1	0	0	0	
3 (3)	1	1	1	0	0	SCC 2
2 (4)	0	1	1	1	1	SCC 3
5 (5)	1	0	0	1	1	

Figure 4 BW

$$
S_1: \begin{cases}
e_1: x_2 + 3x_7 - 2x_{10} - 10 = 0 \\
e_2: \log_{10}(x_2^3 - 17) + x_2^2 - 10 = 0 \\
e_3: -3x_2 - 3x_8 + 15 = 0 \\
e_4: 3x_2 - x_7 - x_{10} = 0 \\
e_5: x_6 + x_{10}^2 + e^{x_{10}-4} - 14 = 0 \\
e_6: 7x_1 - x_5 + \frac{x_2^3}{3} + x_2^2 - 23 = 0 \\
e_7: 2x_2 - 3x_7 + 9 = 0 \\
e_8: x_6 + x_7 - 2 = 0 \\
e_9: 3x_2 - x_6 + x_7 + 2x_{10} - 25 = 0 \\
e_{10}: 2x_2 + x_6 - 2x_7 - x_{10} + 11 = 0 \\
e_{11}: -8x_1 + 15x_6 - 20x_7 - x_{10} + x_2^4 * x_8 + x_2^2 - 14 = 0 \\
e_{12}: x_2^3 - e^{(x_2-3)} - x_3 - 4x_4 - x_5 + x_6 - x_7 + x_9 - x_{10} + 4 = 0
\end{cases}
$$

Figure 5 BW
Click here to download high resolution image

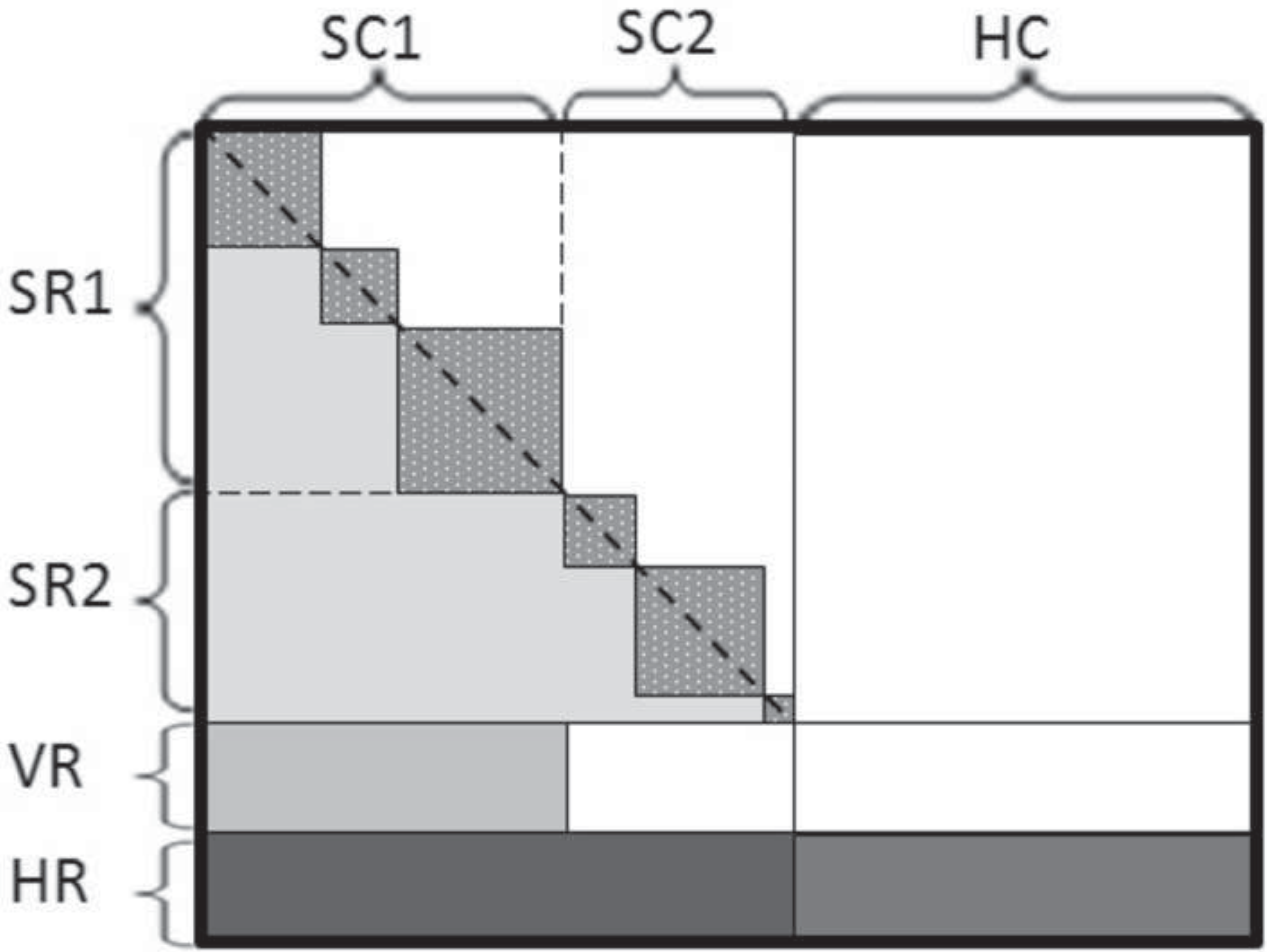


Figure 6 BW
Click here to download high resolution image

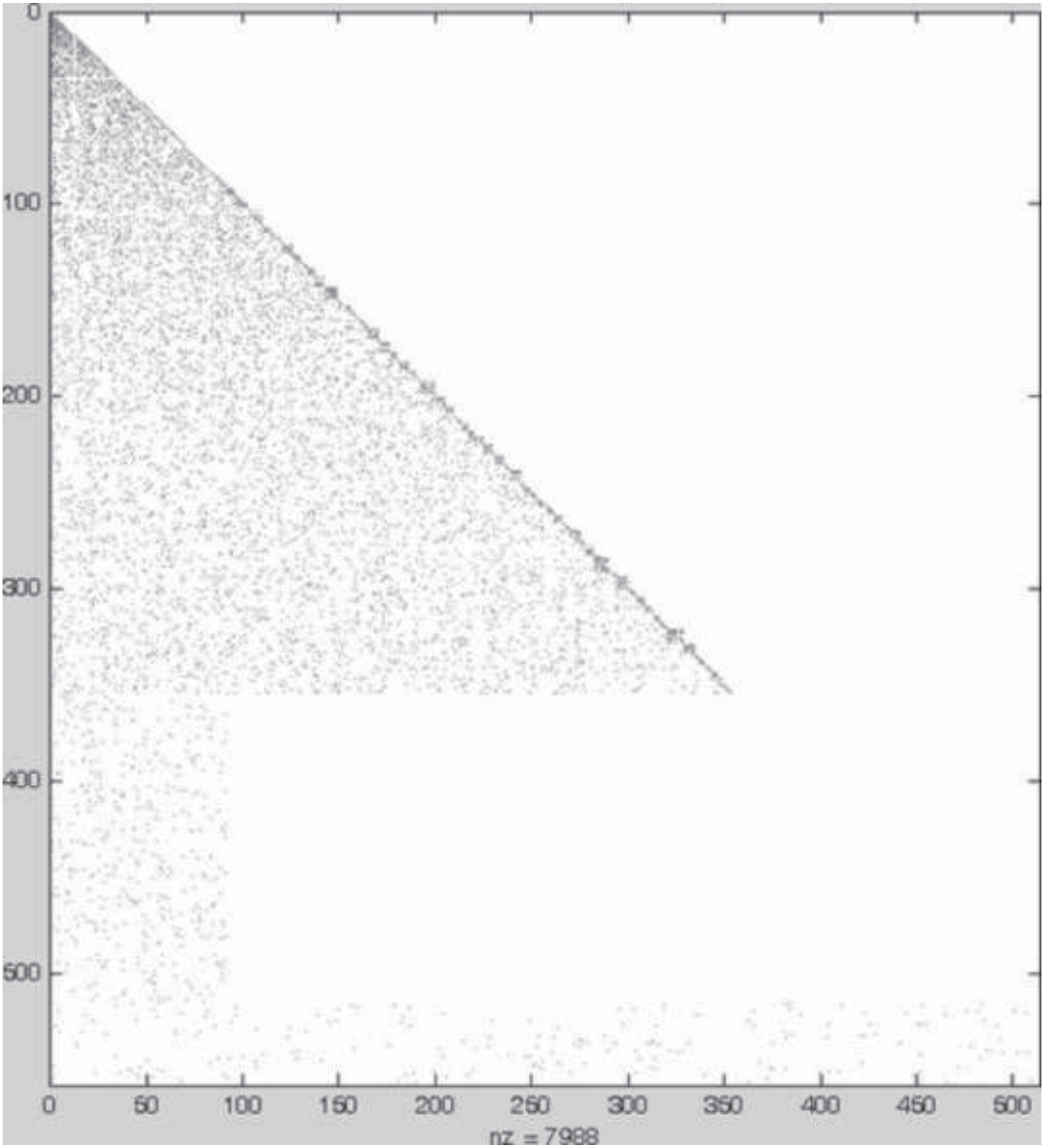


Figure 7 BW

Click here to download high resolution image

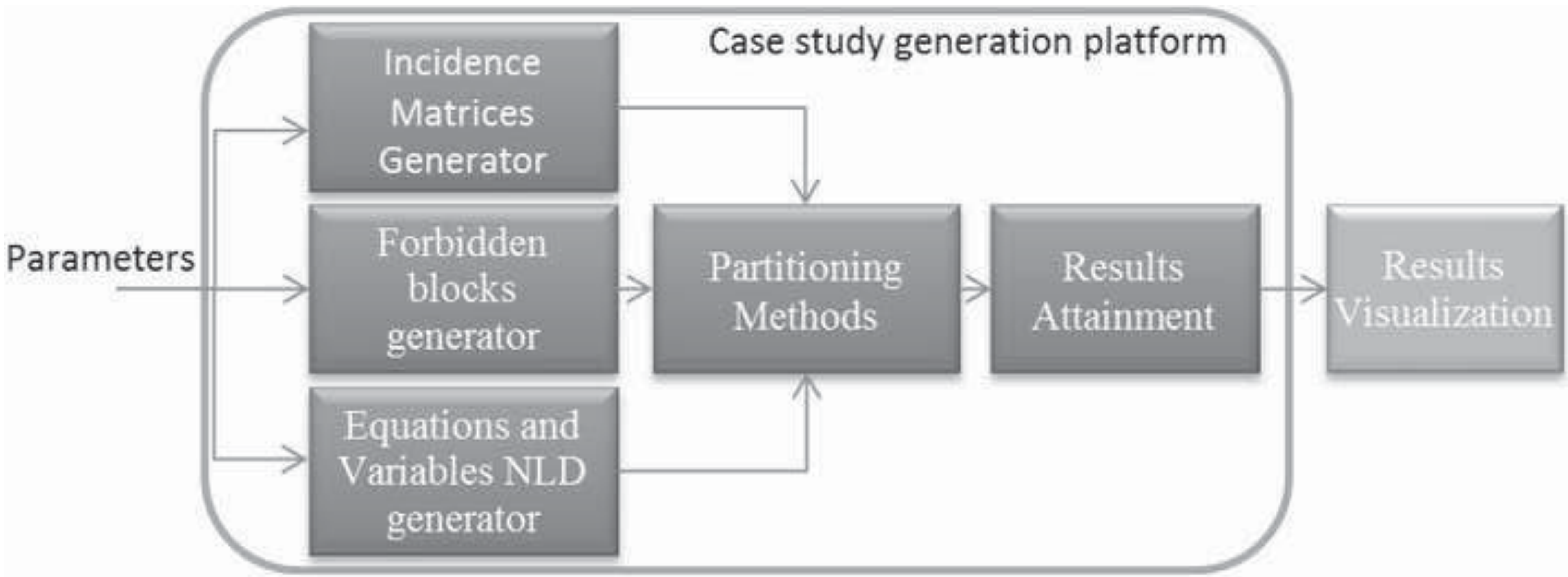


Figure 8 BW

Click here to download high resolution image

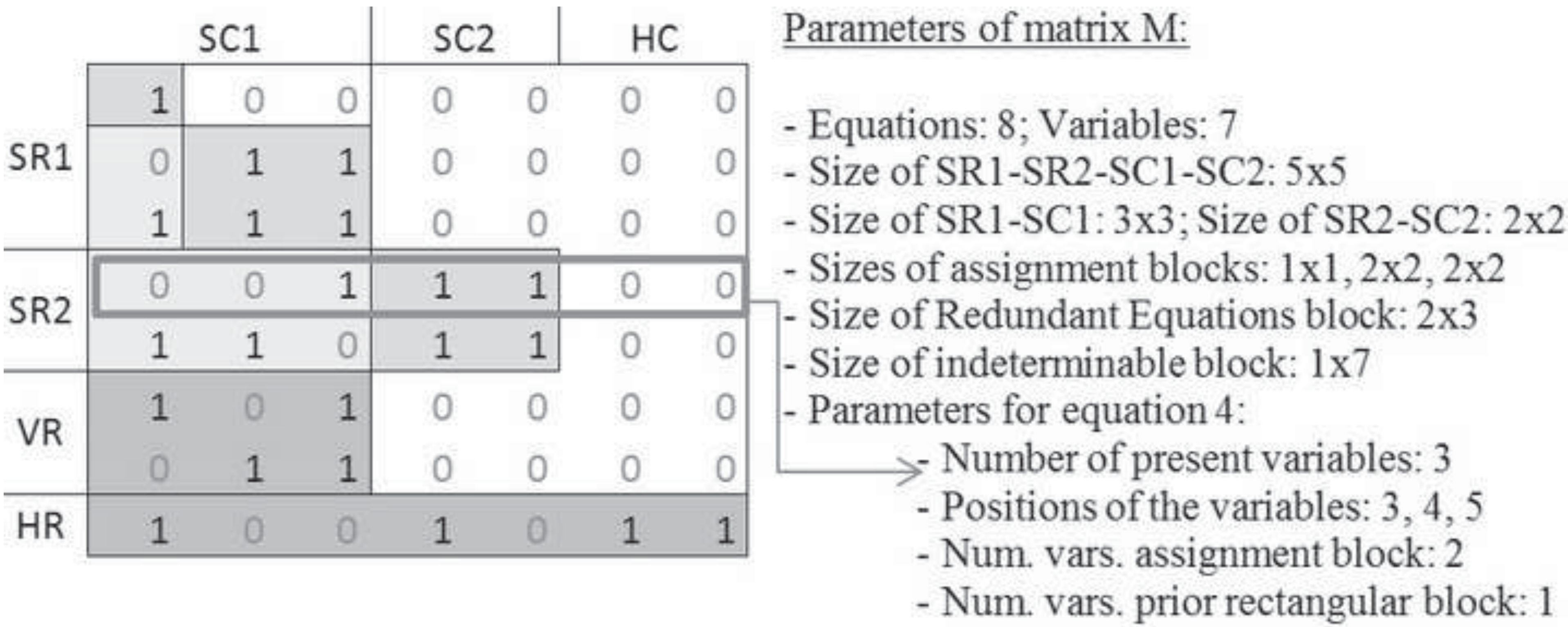


Figure 9 BW

1. *Input: Parameters for incidence matrices building, Q: Number of cases to build*
2. *Repeat Q times:*

2a. *Generate randomly a matrix M in LBTF, based on the input parameters*

2b. *Randomly reorder M rows and columns*

2c. *Generate a set R of constraints for the assignment blocks formation, based on input parameters*

2d. *Define **random** NLD for equations and variables*

2e. *NLD=**0** for a random group of equations and variables, according to the input parameters*

2f. *Run DM, EDM and IEDM for the incidence matrix and the data structures attained*

2g. *Save partitioning methods results*

Figure 10 BW

1. *Input: Params (Statistical parameters for matrix building)*
2. *sizeBlockSR1SR2=RandomInteger(Params->SR1SR2)*
3. *sizeBlockSR1= RandomInteger (Params->SR1)*
4. *sizeBlockSR2= sizeBlockSR1SR2- sizeBlockSR1*
5. *Diagonal(**BlockSR1SR2**)=1*

Assignment Blocks Definition

6. ***Repeat***

6a. *Randomly generate a size for an assignment block*

7. ***Until SR1-SC1 is complete***
8. ***Repeat***

8a. *Randomly generate a size for an assignment block*

9. ***Until SR2-SC2 is complete***
10. ***For each assignment block in SR1-SC1 and SR2-SC2***

10a. ***For each equation (row)***

10a.i. *Set number of variables of the equation and their positions, considering block bounds*

Redundant Equations Block and Indeterminable Block

11. *IndetBlockColumns=Params. Variables-sizeBlockSR1SR2*
12. *IndetBlockRows= RandomInteger (IndetBlockColumns -1)*
13. *RedunBlockColumns=sizeBlockSR1;*
14. *RedunBlockRows=Params.Equations-sizeBlockSR1SR2- IndetBlockRows;*
15. ***For each equation (row) of Redundant Block***

15a. *Set number of variables of the equation and their positions, considering block bounds*

16. ***For each equation (row) of the Indeterminable Block***

16a. *Set number of variables of the equation and their positions, considering that each equation must have at least 2 indeterminable variables*

Figure 11 BW

Click here to download high resolution image

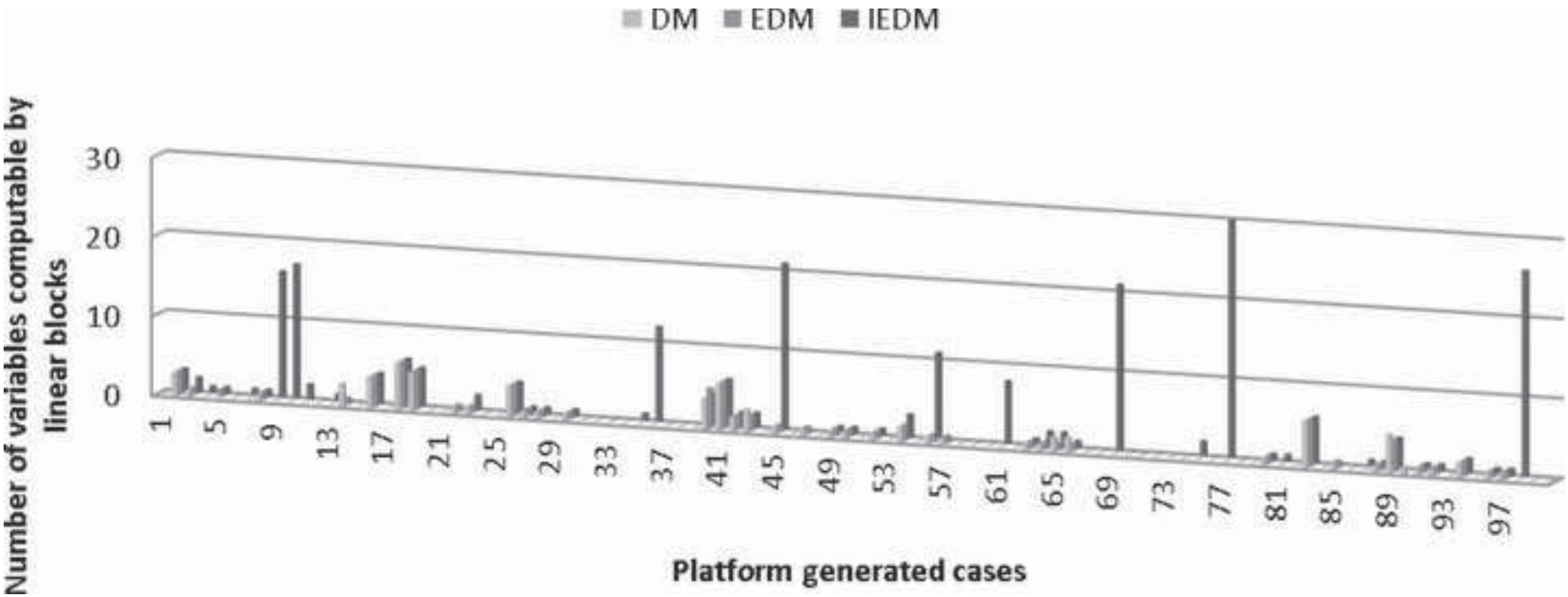


Figure 12 BW
Click here to download high resolution image

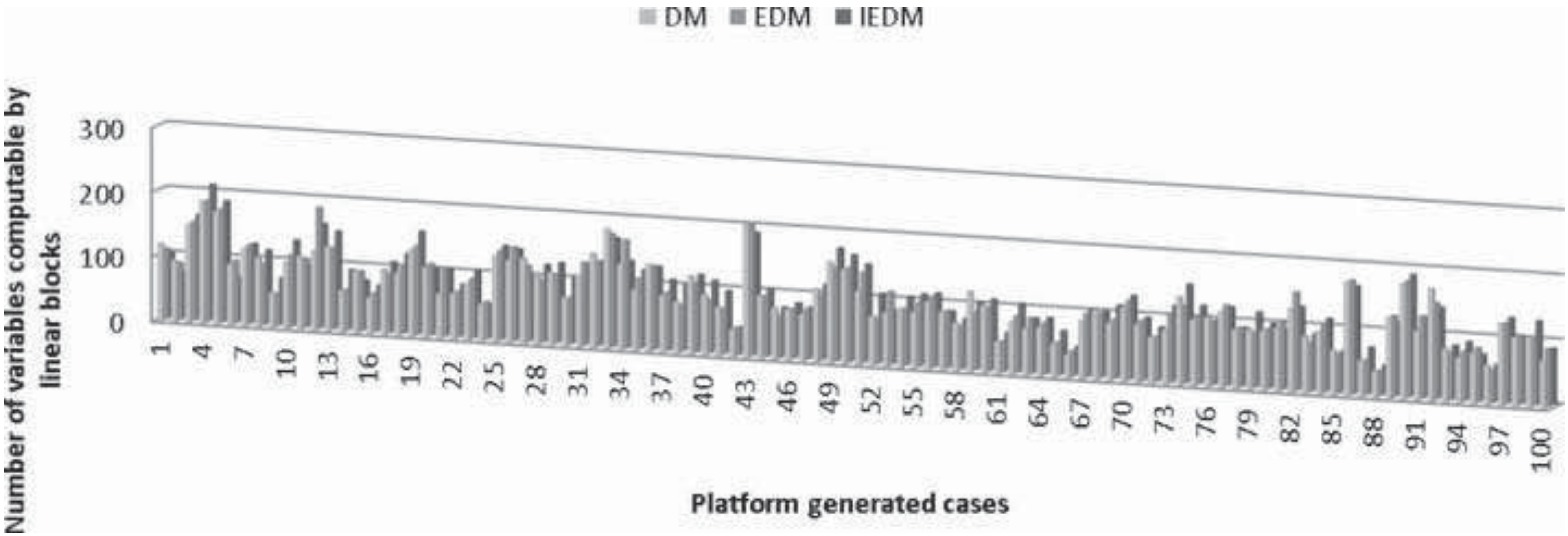


Figure 13 BW
Click here to download high resolution image