A graph theory approach for scenario aggregation for stochastic optimisation

Sergio Medina-González, Ioannis Gkioulekas, Vivek Dua and Lazaros G. Papageorgiou *

Centre for Process Systems Engineering, Department of Chemical Engineering, UCL (University College London), Torrington Place, London WC1E 7JE, UK

*Corresponding author. E-mail address: l.papageorgiou@ucl.ac.uk (L.G. Papageorgiou)

Abstract: The development of fast, robust and reliable computational tools capable of addressing process management under uncertain conditions is an active topic in the current literature, and more precisely for the process systems engineering one. Particularly, scenario reduction strategies have emerged as an alternative to overcome the traditional issues associated with large-scale scenario-based problems. This work proposes a novel and flexible scenario-reduction alternative that integrates data mining, graph theory and community detection concepts to represent the uncertain information as a network and identify the most efficient communities/clusters. The capabilities of the proposed approach were tested by solving a set of two-stage mixed-integer linear programming problems under uncertainty. For comparison and validation purposes, these problems were also solved using the two available methods (*SCENRED* and *OSCAR*). This comparison demonstrates the similar (and in some cases better) quality and accuracy of the proposed approach against the traditional methods. Additionally, the practical advantage of the proposed parameter definition rule is demonstrated as a way to overcome the limitations of the current alternatives (i.e. arbitrary user-defined parameters).

Keywords:
Scenario aggregation, graph theory, community detection and two-stage stochastic programming.

1 Introduction

The explicit consideration of external, unpredictable and very often uncontrollable conditions is a major challenge for efficient process management. In the Process Systems Engineering (PSE) literature these type of problems are addressed through different scenario-based stochastic formulations such as two-stage programming and robust optimisation (Birge and Louveaux, 1997). A common approximation strategy for solving two-stage stochastic programs consists of generating a set of deterministic optimisation problems, by considering a finite number of scenarios representing the uncertainty space (Karuppiah et al., 2010). Note that the performance of this approach depends on generating a sufficiently large set of scenarios ($s \in S$), with an associated probability of occurrence (pr_s) that accurately capture the uncertainty space. In order to systematically cover the whole uncertainty space, a tree structure is usually adopted in which, all the nodes starting from the root node (i.e. known parameter at the first period) are defined as branches for the subsequent periods. Hence, each node has a unique predecessor, but possibly several successors and the branching continues up to the final period at which all the nodes correspond to the total number of scenarios (Growe-Kuka et al., 2003). The number of nodes/scenarios grows exponentially as a function of both, the time periods and the number of successors considered. For example, for a 12 time period problem considering only three levels, there is a need for more than 265,000 scenarios to completely represent only one uncertain parameter.

Despite its good performance, one of the main limitations of the tree-inspired strategy lies in the need to assume a number/level of successors at each stage, which compromise either the problem tractability or the uncertainty space representation. A more simplistic and robust approach consists of randomly generating a large set of scenarios following a known probability distribution, a mean value and a specific standard deviation. Although random selection may be a useful alternative, the need for large datasets to guarantee uncertainty space representation remains as an impractical approach and leads to major computational issues (e.g. intractability) (Sahinidis, 2004). In this line, several decomposition strategies have been proposed to solve problems under uncertainty, including Lagrangian-based (Li and Ierapetritou, 2012) and Benders-based (Chu and You, 2013). Nevertheless, these methods are inapplicable to some problems (such as stochastic capacity expansion) due to the nonconvexities caused by the presence of integer variables in later stages (Ahmed and Sahinidis, 2003). Alternatively, data-driven optimisation strategies emerge to harness the data in an automatic manner and derive smart decision making (Ning and You, 2019). Many methods exist using data-driven along with various uncertainty management optimisation methods such as stochastic programming and distributionally robust optimisation (Shang and You, 2018; Esfahani and Kuhn, 2018), chance constraint (Chen et al., 2018), robust optimisation (Ning and You, 2018a,b) and a few scenario optimisation approaches (Calafiore and Campi, 2005). Even though data-driven methods may be useful to have better control on the quality of the input data, finding the most adequate ratio between the number of scenarios and the accuracy of the uncertainty space representation remains as one of the most critical challenges in uncertainty-aware optimisation problems (Ning and You, 2019). Therefore, the development of fast, accurate and reliable scenario reduction

approaches is needed (Römisch, 2009).

Several scenario reduction algorithms have been developed sharing the same main idea of selecting a subset of scenarios ($s' \in S'$) and give them a probability ($pr_{s'}$) value that allows a good representation of the original dataset. Most of the current scenario reduction approaches seek for the minimal overall distance between the original and the reduced subset of scenarios by strategically splitting the whole set of scenarios. Dupacova and Gröwe-Kuska (2003) applied for the first time this idea, proposing two reduction algorithms known as the forward selection and the backward reduction. These algorithms minimise the global probabilistic distance using the canonical probability metric as a performance criterion by evaluating each scenario individually. Later, this idea was extended by Heitsch and Römisch (2003, 2007, 2009) modifying two main aspects: first, considering the whole set of scenarios at each iteration while evaluating the distances and second, the use of the Fortet-Mourier metric as a distance criterion, leading to a significant improvement in the computational performance (i.e. accuracy and computational time). Recently, Chen and Yan (2018b) proposed a scenario tree reduction formulation to cluster nodes with the same parent node into a smaller number of nodes. More recently, Silvente et al. (2019) proposed *RedOpt* which is a scenario reduction technique inspired by the Branch and Bound method that progressively evaluates the branches using a sensitivity analysis. The good performance of these algorithms not only has justified/boosted their application to address chance-constrained and mixed integer-two-stage stochastic programming problems but also has enlarged the scope of scenario reduction methods to address various practical application such as energy production Xu et al. (2015), chemical processes Karuppiah et al. (2010) and the pharmaceutical industry (Henrion et al., 2008, 2009).

The insights of scenario reduction methods have led to the development of several clustering-based scenario reduction tools, such as *SCENRED* (GAMS documentation, 2019) and *OSCAR* (Li and Floudas, 2016) which are widely used. Contrary to OSCAR, SCENERED is used extensively by GAMS users mainly since it is already included in its available options and therefore no pre-processing steps are required for its use. Both techniques use *k-means* approach as the core algorithm since it consists of a simple iterative process that accounts for the global distance to assign samples into k clusters (with k being user-defined) and then averages all the samples to create the new cluster centres (Hastie et al., 2008). Despite their use in a wide range of applications and proved computational benefits (Lima et al., 2018; Zeballos et al., 2018; Chen and Yan, 2018a; Medina-González et al., 2018a), the main limitation of SCENRED and OSCAR is their loss of accuracy and/or time effectiveness when the original set is significantly large ($> 4,000$) or when the original dataset represents a low dimensional problem (Li and Li, 2016). Another important limitation of these methods is the lack of a rule/strategy to determine the reduced set size (i.e. highly subjective approach). Considering the above limitations and knowing that some authors have acknowledged that the proximity of each scenario to the cluster centre might not be the best clustering driving force (Bhagat et al., 2016), there is a need to improve the above approaches following a strategy that promotes a deep and step-wise analysis of the uncertain parameters.

A promising alternative lies in exploiting all the indirect information from the original uncertainty dataset in the form of pairwise correlation values. Later, data mining can

be used for the exhaustive evaluation/analysis of such data, and by integrating it with graph theory, a meaningful data arrangement/visualisation can be obtained as a network (Ning and You, 2019; You et al., 2018). Individually, these methods have been employed to address problems from different fields, including computer science and engineering. For example, graph theory has been applied to a wide range of data management applications such as worldwide web analysis, collaboration networks and understanding complex biological systems (Newman, 2018). On the other hand, machine learning was successfully used for the efficient process management of large manufacturing processes (Kusiak, 2006; Monostori and Viharos, 2001). Nevertheless, to the best of the authors' knowledge, they have never been combined to address PSE challenges, neither have been integrated into a scenario reduction framework. Note that being able to translate the uncertainty dataset into a well-constructed network opens the opportunity to exploit various features (such as connectivity, node influence/importance and more importantly its inherent community structure) to cluster the dataset, keeping an accurate uncertainty space representation regardless of the amount, dimensionality, behaviour and complexity of the original information.

This paper proposes a novel and flexible scenario-reduction alternative that integrates data mining, graph theory and community detection concepts. A brief background is presented in the following subsections, while the particular tools composing the solution strategy are presented in Section 2 along with the obtained results (Section 3). Finally, the main outcomes of this paper are summarised in Section 4.

2 The SCANCODE algorithm

The proposed strategy named *SCANCODE* (that stands for *SC*enario *A*gregatio*N* and *CO*mmunity *DE*tection) is a novel scenario reduction approach that expresses dataset through a network representation. This strategy consists of four main steps/parts as shown in Fig.1: Parameter initialisation/declaration, network construction, cluster generation and centroid identification. In the following subsections, each step is explained in detail and finally, the overall algorithm is summarised.

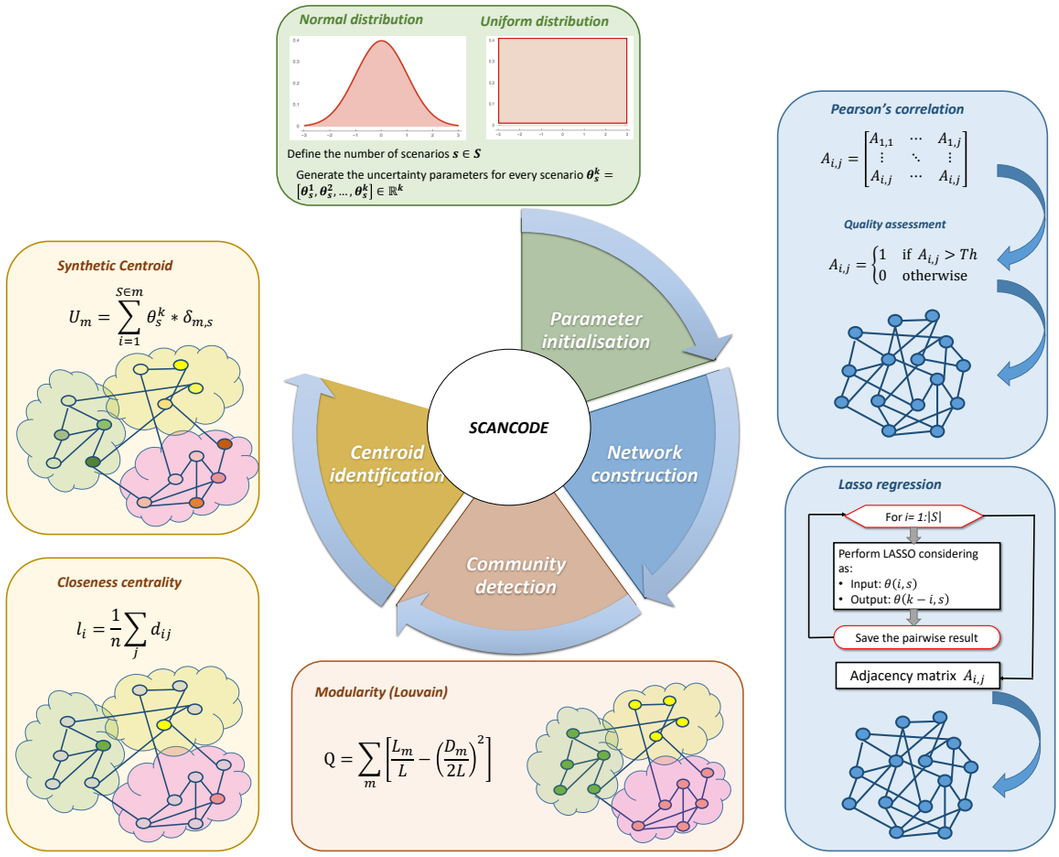

Figure 1: Overview of the proposed methodology.

Once the set of scenarios has been generated, the general *SCANCODE* algorithm is applied as follows:

SCANCODE algorithm

Step 1.	Network generation. Create the adjacency matrix by calculating the pairwise relationship of all vertices.
Step 2.	Community detection. Identify the clusters in the network by applying a suitable community detection algorithm.
Step 3.	Identify the reduced set of scenarios, based on the clustered network of the previous step.

At its core, SCANCODE creates a complex network that represents the relation or correlation between all the pairs of points (e.g. scenarios). This complex network is then clustered into groups by utilising available community detection methods.

Traditional methodologies include graph partitioning, hierarchical and spectral clustering, and also divisive algorithms such as the Girvan-Newman algorithm. Modularity-based methods maximise the modularity metric (Newman, 2003), which has become an essential element of many clustering methods. One such methodology is the OptMod algorithm (Xu et al., 2007). This algorithm uses mixed-integer optimisation in order to partition the graph into m communities while maximising the modularity metric. The algorithm achieves very good modularity values on a selection of benchmark datasets, but it is computationally expensive. In contrast, greedy optimisation algorithms have

the advantage of computational speed and handling of large networks. One of the most popular methods is called Louvain, which is a two-step algorithm that maximises the modularity metric. For a given network, the first step assigns nodes into clusters increasing the modularity value whereas the second step creates a new network where each node represents a cluster from the previous step. These two steps are iterated until no further improvement is possible (Blondel et al., 2008).

Finally, within each identified cluster, the most representative nodes are defined as the cluster centroids. The nodes' importance can be defined following different user-defined criteria (based on the network centrality, connectivity, etc). Regardless of the criteria employed, the centroid/central node keeps its original values, thus, the final subset of scenarios consists of those nodes in the original set with the greatest relevance. Alternatively, a synthetic/artificial centroid may be derived for each cluster.

The SCANCODE algorithm is a methodology that allows the user to create a graph and choose the desired clustering algorithm in order to identify a reduced set of representative scenarios. This section describes the methodology that was followed for steps (2), (3) and (4) of the proposed algorithm.

2.1 Parameter initialisation

A representative set of scenarios has to be created so as to properly capture the uncertainty space. Even though various scenario generation/construction approaches may be used, in this strategy a random selection is employed. Essentially, a user-defined number of scenarios is determined and a value for each dimension of the uncertain parameter is randomly selected within a known probabilistic distribution. An equal probability of occurrence was assumed for all the scenarios calculated as follows:

$$pr_s = \frac{1}{|S|}$$

where s represents the scenario index and $|S|$ accounts for the total number of scenarios.

Note that this step belongs to scenario generation rather than to scenario reduction algorithm. However, its inclusion in this paper is to illustrate that the proposed approach can be used regardless of the scenario generation method used.

2.2 Network construction

The creation of a network that captures the pairwise relationship of the scenarios is vital. In this work, two different methodologies have been used to generate a network. The first method uses the *Pearson's correlation coefficient* whereas the second one uses *LASSO* regression. A brief description of the idea behind these methods is presented below but for more information, we refer the reader to appendices A.4 and A.5 respectively.

2.2.1 Pearson's correlation coefficient

This metric is commonly used to identify the linear correlation between variables in a dataset. In this work, however, the metric is used to identify the correlation of the scenarios, based on the values of a given set of uncertain parameters (P_m). The product of applying that coefficient is a square symmetric matrix with values in the range [-1,1] (Known as Adjacency matrix, $A_{i,j}$).

For the sake of simplicity, it is desirable to generate an unweighted and undirected network without self-loops. The first step is to take the absolute value of all the elements, leading to a matrix in the range of [0,1], in which every non-zero value represents a connection between two particular scenarios. Additionally, the elements of the main diagonal are set to 0 to remove self-loops in the network.

Applying the correlation metric and following the steps described above could potentially generate a very dense network since most of the scenarios have some degree of correlation between them. To address this issue while guaranteeing an unweighted network, the following measures are applied:

- A threshold value (Th) is defined for the *Pearson's correlation matrix* based on the decision-makers interests. If a value is lower than the Th, then the correlation between those two scenarios is rejected for the final network, and the value of the element is substituted with 0.
- All the remaining non-zero elements of the adjacency matrix are set to the value of 1, leading to an unweighted network. The final product is a matrix in the range {0,1}.

Note that the threshold value affects the resulting network. Even though few approaches have been proposed to determine efficient initialization values, they have been heavily criticised due to their lack of accuracy (Celebi et al., 2013). Therefore, a sensitivity analysis is suggested in this paper to identify those values that satisfy the decision-maker criteria.

2.2.2 LASSO regression

Lasso regression is a machine learning method that can be used for variable selection and regularisation on a set of supervised data. In this work, there is no output variable in the data but only uncertain parameters, meaning that the available dataset is unsupervised. Fig. 2 below illustrates the procedure used to apply Lasso in this dataset and generate a network.

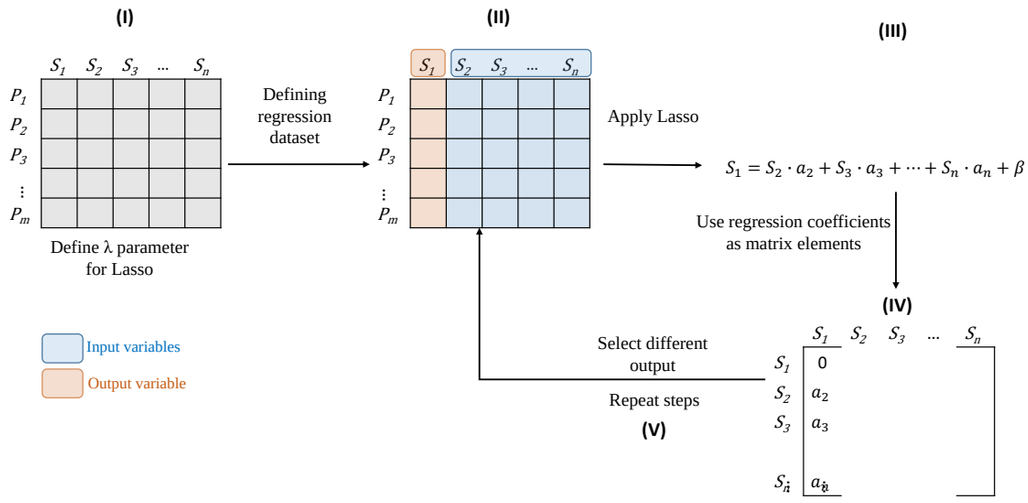

Figure 2: Network generation using Lasso

The five steps included in this approach can be summarised as follows:

- Step I. To increase interpretability for this approach, the first step defines the dataset as a matrix where rows represent the uncertain parameters and columns represent the number of scenarios. At this step, the penalty parameter (λ) for Lasso is defined. This value will remain the same for all further iterations. More details about Lasso regression and the definition of λ can be found in Appendix A.5.

- Step II. Define one scenario as the output variable and the rest as input variables for Lasso regression. This step is necessary to convert the data into a supervised dataset. Essentially, the aim is to try to predict the values of the uncertain parameters of a single scenario based on the values of the remaining ones.

- Step III. Apply Lasso regression to the dataset generated in the previous step. Since Lasso is a shrinkage method, meaning that it penalises the regression coefficients by using a parameter λ, some regression coefficients will be forced to the value of 0. This process of feature selection is vital for the next step.

- Step IV. Using the results of step III the adjacency matrix is filled, where each element has the value of the corresponding regression coefficient. Note that in order to avoid self-loops the valued of the main diagonal are set to 0.

- Stage V. Repeat steps 2 to 4 changing at each iteration the output variable until all the scenarios have been iterated.

When this procedure is finished, the result is a non-symmetric square matrix. Once again for simplicity reasons, every non-zero element is set to the value of 1 in order to create an unweighted network. However, in order to ensure that the network will not be directed the matrix should be symmetric, thus, a final step is necessary. Assuming that i and j represent the nodes in the network, we apply the following condition.

$$if \quad A_{i,j} = 1 \quad or \quad A_{j,i} = 1$$
$$A_{i,j} = A_{j,i} = 1 \qquad \forall\ i,\ j,\ i \neq j$$

Remarkably, the Th and λ value described in this section affects the network density, and ultimately the number of clusters in the reduced set. Such an issue will be addressed in the following subsection.

2.3 Community detection (Clustering)

A community is a group of *nodes* densely connected between them and sparsely (or not at all) connected with other communities. Thus, in this part of the algorithm, a meaningful community partition is identified from the network obtained in the previous step by optimising the *modularity* metric described before.

The proposed approach employs the well-known Louvain algorithm to solve the community detection problem (Blondel et al., 2008) even though there are several alternative methods including Girvan-Newman and Tyler-Wilkinson (Javed et al., 2018), which as described in the previous subsection is a two-step algorithm that maximises the modularity metric.

According to Newman (2003), after applying a modularity-based clustering algorithm, if the resulting modularity value is above 0.3, then the set of identified communities is thought to be well structured and well defined. Since the Th and λ value affects the network density, any value leading to a modularity value above 0.3 is considered good enough to solve the problem. The above implies that the proposed strategy can provide the minimum number of clusters that produce an acceptable representative uncertainty space. The above represents a practical advantage over the current clustering-based approaches used in stochastic programming.

To determine/initialise the Th and/or λ values different rules have been used. For Pearson's correlation coefficient (Th) we follow an iterative process increasing the Th and calculating its associate modularity value. For any Th with a modularity value greater than 0.3, the decision-maker might use a trade-off between the modularity value and the number of "clusters". For Lasso regression (λ) an initial lambda value may be calculated as explained in Appendix A.5.1, from which the decision maker can explore around.

2.4 Centroid identification

Once the communities are defined, the most important/relevant centroid can be identified for each community by selecting a *node* based on a centrality criterion (i.e. *Degree centrality* and *Closeness centrality*)(Newman, 2018). These metrics depend on graph density and connectivity, but ultimately select a single node as the most central one. Therefore, another option could be adopted that takes into account the information from all the scenarios that are present in a specific cluster. This new synthetic centroid, will be a weighted average of the *in-degree* in order to capture the influence/importance of each node within a cluster. In this context, in-degree is the total degree of all the node in a cluster but only considering the connection within that cluster. For more information about degree centrality the reader is referred to appendix A.1

3 Case studies

Speeding-up the optimisation of two-stage stochastic problems is one of the many potential applications of SCANCODE as scenario reduction method. Hence, in this section SCANCODE is tested using three examples with different features. The details of each case study are presented in their respective subsections. For comparison purposes and quality assessment, the optimal solutions associated with SCANCODE were contrasted with those obtained using SCENRED and OSCAR.

Note that the main difference between OSCAR/SCENRED and SCANCODE is that the first ones are clustering-based approaches while the later one is a graph-clustering-based approach. Essentially, while OSCAR/SCENRED simply apply clustering methods to split a dataset, SCANCODE translates the dataset into a network and later identifies the module-based structure (that are used as clusters).

The MILP models were implemented in GAMS 24.7 and solved using CEPLEX 12.6.3 to a relative optimality gap of 0% for the first two examples and 2% for the third one.

3.1 Validation strategy

Typically, scenario reduction approaches assess their performance by comparing their expected objectives. However, this value does not necessarily capture the performance of the solution over the whole uncertainty space. Thus, in the paper, we compared simultaneously three different values, expected (EP), post-process (PP) and full-space performance (FS). Specifically, EP represents the solution of the problem using exclusively the subset of scenarios, PP is the solution of the relaxed version of the stochastic problem (i.e. fixing the 1^{st} stage decisions obtained in EP) for the entire superset of scenarios, and finally, the FS performance represents the solution of the two-stage stochastic problem for the entire superset of scenarios. Ultimately, the method that achieves the smallest gap between them is considered the most accurate/reliable one.

In addition to the objective performance, the computational one is also compared. Note that the computational time reported consists of the time required to produce the reduced set of scenarios plus the one needed to obtain the expected performance. Remarkably, the computational time to solve the PP problem was not considered for two main reasons. First, the PP time is similar (if not the same) regardless of the scenario reduction method used since it consists of the LP optimisation after fixing the binary variables. Second, the ultimate objective of any scenario reduction algorithm is to obtain the most representative EP for the uncertainty space. Finally, PP values in this paper were used only to validate the quality of the solution obtained with the reduced set of scenarios based on the entire uncertainty space.

3.2 Bio-based energy production supply chain

In this case study, the objective is to maximise the economic benefit of a biomass-based energy SC system through the optimisation of its design and operations (See Fig. 3).

The system consists of nine districts, acting simultaneously as biomass suppliers, energy generation and market sites. 40 different biomass states were considered across the entire entities, six available technologies and 79 activities were considered across a monthly discretised year horizon. Cassava Rhizome (CR) was used as the raw material for energy production, while its availability and the total energy demand were considered as the uncertainty sources. Notice that they were defined only in the first and last raw material states respectively while the remaining 38 states are calculated as a function of the operations. Additionally, it was assumed that the raw material is only produced during five months which corresponds to the length of the CR production season. Consequently, 108 uncertain parameters are assumed for demand while 45 for availability. Following the same approach as described in Medina-González et al. (2017), 50 randomly generated scenarios with the same probability of occurrence were used to represent the process uncertainty for each parameter. Compared to the number of uncertain parameters, the superset of 50 scenarios is hardly representative, however, the problem cannot be solved directly with two-stage stochastic programming for larger superset sizes due to computational limitations. Regardless of the above issues, this case study was included in the paper to evaluate the performance of the proposed approach to solve problems of different complexities and under various conditions, including large/low number of scenarios, different probabilistic distribution shapes, and large-low dimensionality.

For this case study, a normal distribution with the average values presented in Table 1 and a standard deviation of 30% was used. Since the scope of the paper is to describe and test the proposed scenario-reduction approach rather than modelling the problem, the complete mathematical formulation of the MILP problem is presented in the supplementary material file while the main parameters can be found in Medina-González et al. (2017).

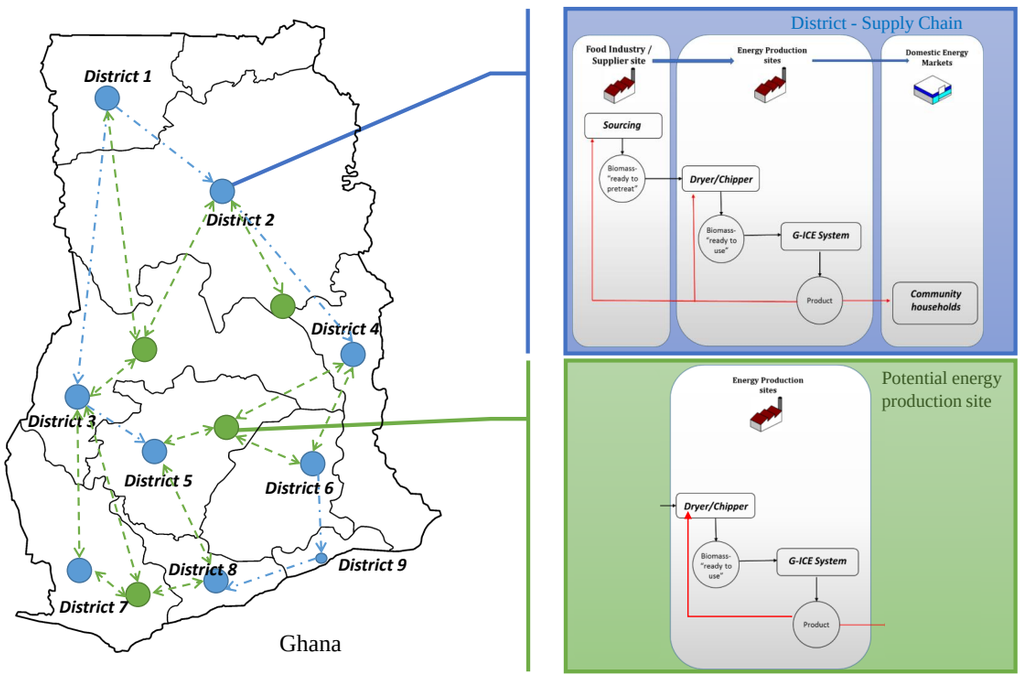

Figure 3: Bio-based network superstructure for case study 1

District	1	2	3	4	5	6	7	8	9
Demand (MJ)	1,942	4,055	15,250	19,363	2,684	3,198	913	3,884	4,169
Availability (kg)	12.8	24.4	81.1	122.2	16.2	22.1	5.3	21.1	28.2

Table 1: Mean values for uncertain conditions at each district.

3.2.1 Network and clustering results

Two different adjacency matrices were obtained by applying *Pearson's correlation coefficient* for the first one and *Lasso regression* for the other. A coloured illustration of the Pearson's pairwise relationship is displayed in Fig. 4, which can assist in the definition of the different threshold values. For this particular case, *Th=0.90, Th=0.905, Th=0.91, Th=0.92* and *Th=0.93* were used.

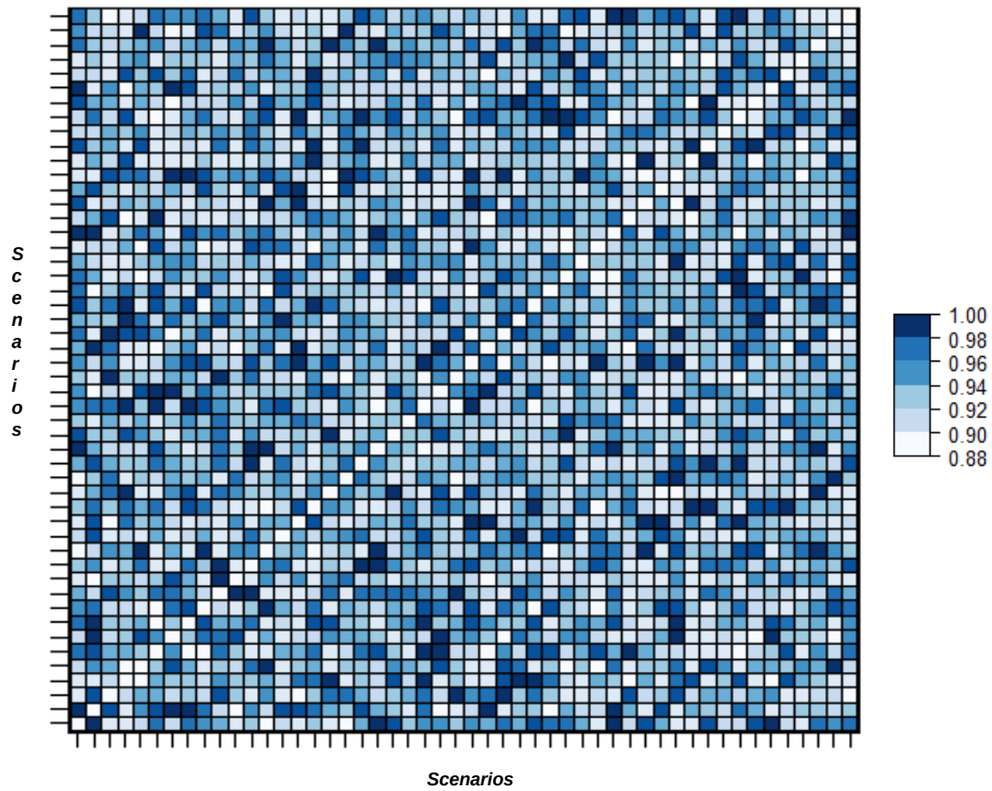

Figure 4: Pairwise relationship level for Pearson's correlation

For the case of Lasso regression, both the pairwise relationship generation and the quality filter are performed simultaneously as described in section 2.2.2. Thus, five equidistant penalty parameters ranging between ± 30% of the initial value have been considered (λ=0.029, λ=0.035, λ=0.040, λ=0.046 and λ=0.057).

Finally, using the filtered adjacency matrices and following the process described in section 2, the associated networks as well as the communities within them were obtained. A total of 5, 5, 7, 19 and 34 communities were identified for Pearson-based networks while 6, 6, 8, 18 and 40 of them for Lasso-based networks (Fig. 5).

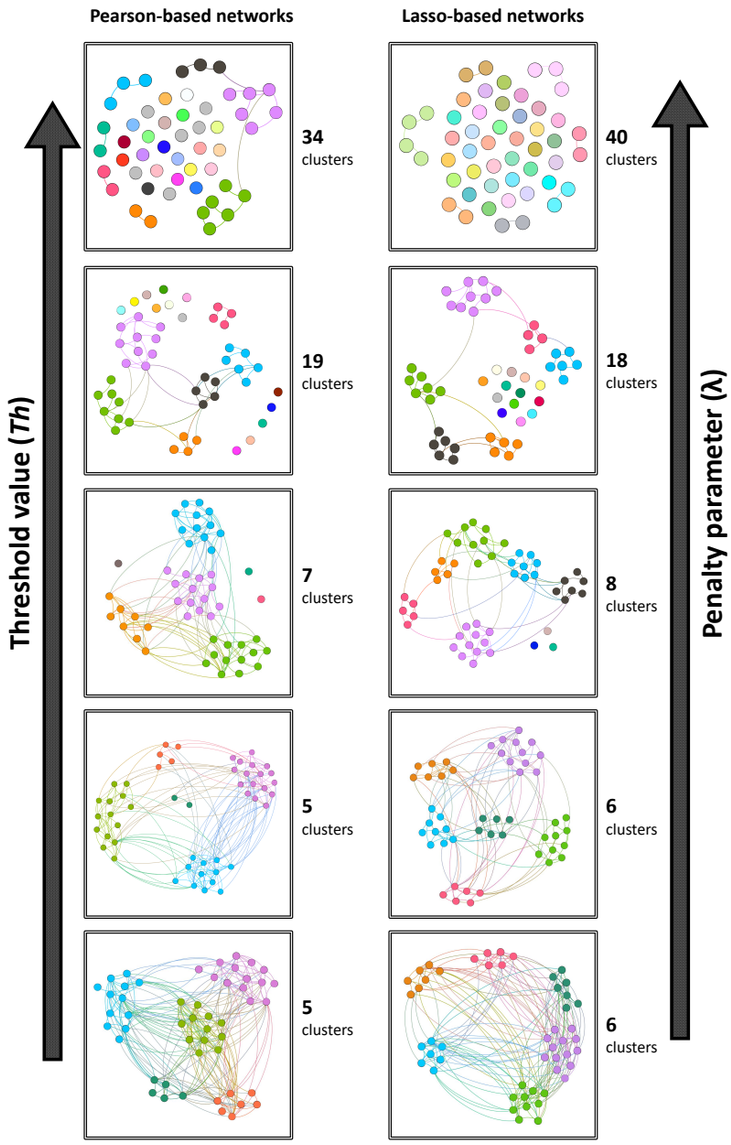

Figure 5: Networks and Clusters for different threshold values

Analysing the obtained networks (Fig. 5), it is clear that their density is inversely proportional to the Th/λ value no matter which method was used to construct the pairwise relationships. In both cases, the larger the parameter value, the larger the number of scenarios considered as isolated clusters. Conversely, for small Th, a highly dense network is obtained which might compromise the cluster centroids definition. It is worth noting that in general, for Lasso-based networks there are fewer connections between clusters as well as isolated points which might suggest a better and neater clustering.

In order to confirm the undesirable/impractical properties in the reduced set of scenarios for the extreme threshold points, Th=0.85 and Th=0.95 were briefly studied leading to 3 and 49 final clusters respectively (See Fig. 6).

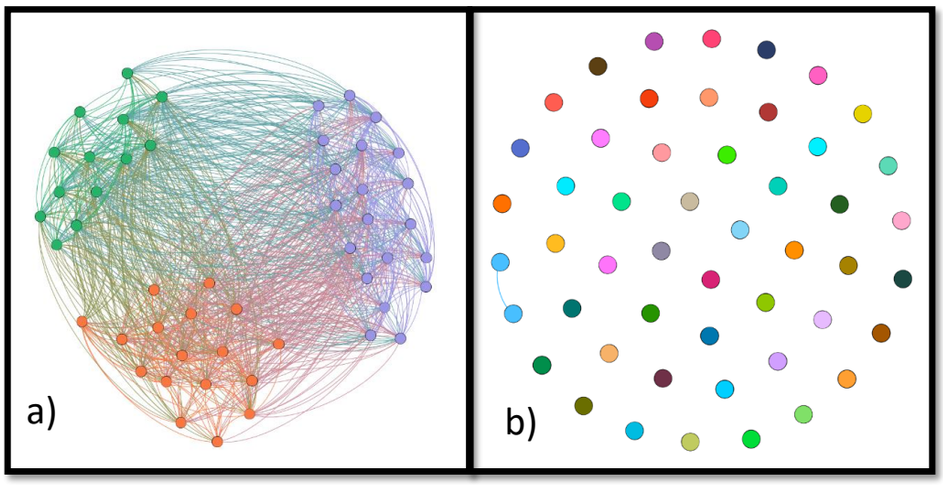

Figure 6: Networks associated to extreme threshold values a) Lower bound ($Th = 0.85$) and b) Upper bound ($Th = 0.95$)

Even though the same general behaviour was found for both of the pairwise relationship method used, their effect over the final network was assessed. As briefly mentioned before, Lasso-based networks lead to better-structured communities which can be confirmed by comparing the modularity value achieved as a function of the cluster size (Fig.7).

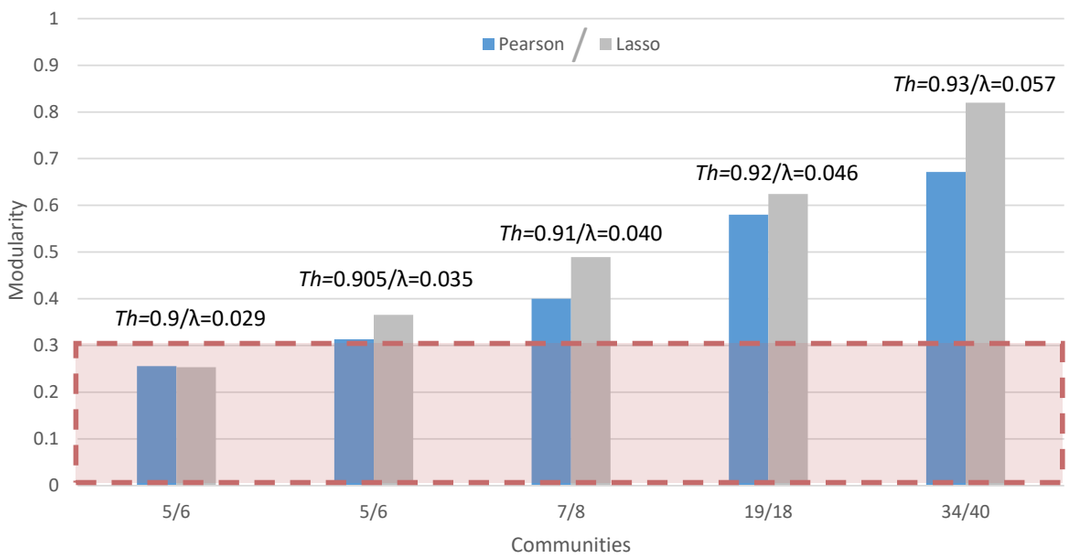

Figure 7: Relation between modularity value and clusters size. The shaded box represents the rule that modularity values should be above 0.3 (Newman, 2003)

The above proves that for a similar number of clusters, the resulting community arrangement is better in Lasso-based networks than in Pearson-based ones. A clear example of this behaviour can be found by looking at the fourth column in Fig. 7 (19/18 clusters) in which for a lower number of communities in Lasso-based networks, a better modularity value is achieved.

After identifying the clusters within the network, the most representative nodes are isolated for each one. Particularly, two different approaches are considered to evaluate the effect of the cluster centroid over the optimisation process. For the first approach, the node with the highest *closeness centrality* value is selected as the principal node,

while for the second one, a synthetic centroid was derived by calculating the weighted average value for the uncertain parameters considering the node's *in-degree* ratio as the importance metric at each cluster. Finally, it was considered that in both cases, the cluster/centroid probability is modelled by the aggregation of the original probabilities of all the elements forming the cluster.

3.2.2 Case 1: Process optimisation results

The resulting reduced sets of scenarios are then used as input data for the two-stage stochastic problem illustrating how the proposed approach affects the solution. OSCAR and SCENRED are used too to solve the same problem. To ensure a fair results comparison, both, OSCAR and SCENRED, are set to generate the same number of scenarios as SCANCODE. Note that the derivation of synthetic centroids for OSCAR and SCENRED is not possible since there is a lack of detailed information about the elements within each cluster. The obtained results are displayed in Table 2.

Table 2: Results for the optimisation of the Bio-based energy production system.

	Clusters	***Modularity***	***ExpProfit*[a] *(EP)***	***PostProcess*[a] *(PP)***	***CPU seconds***
Full-space (FS)	50		340,653		1,630.28
			Pearson-based network		
Th			*Closeness centroid*		
0.900	5	0.2559	357,175.9	340,652.9	31.88
0.905	5	0.3134	345,859.3	340,652.8	29.05
0.910	7	0.4002	356,660.9	340,652.9	43.27
0.920	19	0.5804	350,611.9	340,652.9	159.27
0.930	34	0.6713	339,670.0	340,652.8	380.08
Th			*Synthetic Centroid-Pearson*		
0.900	5	0.2559	347,607.1	340,652.9	29.27
0.905	5	0.3134	347,891.4	340,652.9	30.52
0.910	7	0.4002	347,866.8	340,652.9	42.89
0.920	19	0.5804	344,427.5	340,652.9	163.33
0.930	34	0.6713	341,357.5	340,652.9	313.39
			Lasso-based network		
λ			*Closeness centroid*		
0.029	6	0.2537	348,646.0	340,652.8	34.38
0.035	6	0.3659	336,301.8	340,652.9	37.14
0.040	8	0.4895	343,287.6	340,652.9	51.63
0.046	18	0.6244	340,513.8	340,652.9	139.74
0.057	40	0.8200	339,235.1	340,652.8	452.36
λ			*Synthetic Centroid-Lasso*		
0.029	6	0.2537	348,413.5	340,652.8	33.05
0.035	6	0.3659	347,697.9	340,652.9	36.70
0.040	8	0.4895	347,247.4	340,652.9	51.59
0.046	18	0.6244	345,294.1	340,652.9	141.27
0.057	40	0.8200	341,562.1	340,652.9	450.23
			SCENRED		
–	6	–	331,011.2	340,652.9	38.30
–	6	–	331,011.2	340,652.9	38.30
–	8	–	331,101.8	340,652.9	53.69
–	18	–	332,808.1	340,652.9	168.75
–	40	–	338,287.0	340,652.9	2,040.35
			OSCAR		
–	6	–	347,538.7	340,652.8	32.98
–	6	–	347,538.7	340,652.8	32.16
–	8	–	336,037.0	340,652.8	52.40
–	18	–	345,811.9	340,652.9	143.89
–	40	–	344,541.2	340,652.8	513.27

[a] (€)

From Table 2 it can be seen that there is a significant variation in the *EP* values, which is anticipated since the scenarios in the reduced set may be (and very often are) different for each strategy. From these results, two main elements should be highlighted. First, the computational effort needed to solve the problem using SCANCODE is slightly lower than using OSCAR or SCENRED; Secondly, in SCANCODE different Th/λ values may lead to reduced sets of scenarios with the same size but different community arrangements, modularity values and ultimately optimal decision values (See Th=0.9 and Th=0.905 as well as λ=0.029 and λ=0.035). The second can be explained if we look at the core difference between SCANCODE and OSCAR/SCENRED. For the last ones, the user-intervention lies on directly defining the reduced set size (user-defined approach) while in SCANCODE, the input data is represented as a network using a parameter defined by the user (user-induced approach). For example, for any alter-

ation in the user intervention, OSCAR/SCENRED will strictly lead to a different size in the reduced sets of scenarios, while SCANCODE may lead to reduced sets with the same size but different specific scenario allocations. This is because in SCANCODE the user indirectly determines the scenario connections (network) and using this information/dataset the rest of the algorithm determines the best possible uncertainty space representation at the lowest set size. Hence, the representation may be improved with the same number of clusters if more significant "connections" between scenarios are used.

The above suggests that SCANCODE improves OSCAR/SCENRED capabilities from a practical perspective since the defined Th/λ may be evaluated at an early-stage (before solving the stochastic problem). Additionally, SCANCODE always identify the minimum number of scenarios that most accurately represents the uncertainty space for a defined Th/λ. Even though it was not explored in this paper, these data can be further exploited for in-cluster analysis to improve the final clustering quality.

Despite the fact that slightly different 1^{st} stage decisions were obtained for different methods, Table 2 shows that the difference between *PP* and *FS* can be neglected ($<$ 1.0x10^{-6}%). The above suggests that different decisions may lead to similar solutions, but more importantly, the proposed strategy is at least as good as the conventional methods to reproduce the global process performance under "limited data availability". Nevertheless, it is worth to highlight that the *EP* obtained using SCANCODE is closer to the *FS* one which is relevant for real-life problems since the post-process validation is not a common practice due to time limitations. Hence, any scenario reduction strategy represents a useful decision-making tool only after closing the difference between *EP* and *FS*. These deviations are visually illustrated in Fig. 8.

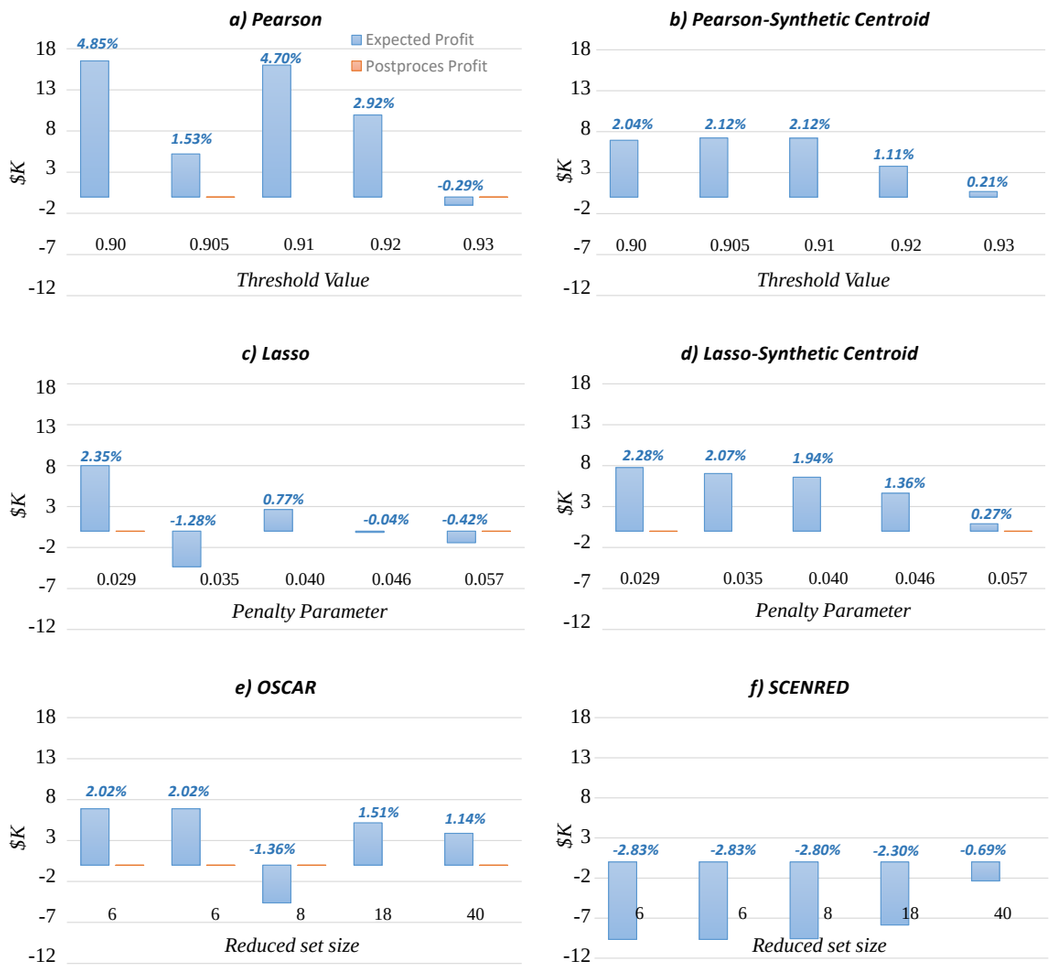

Figure 8: Deviation of the optimal solutions for 5 different levels of reduction for the bio-based energy network problem.

For the case of *EP* vs *FS*, Fig. 8*a* demonstrates that for Pearson-based networks the largest gaps are obtained (up to 4.8% and 2.7% on average) being the less reliable decision-making approach for this example. On the contrary, Lasso-based networks (Fig. 8*c*) achieves the global lowest deviations with a maximum gap of 2.1% proving that Lasso-based networks are better at closing the gap between *EP* or *PP* vs *FS*. The above supports the idea of using sophisticated regression/correlation techniques while constructing the pairwise relationships to better represent the uncertainty space regardless of the reduction level.

Despite the above results being obtained by using as cluster centroids the nodes with the highest closeness centrality value, the effect of using a synthetic centroid instead was also explored. Particularly, for the case of Pearson-based networks, a significant reduction in the gaps was achieved with a single exception of *Th=0.905*, while for Lasso-based networks, the gaps were not further closed. Remarkably, a similar behaviour was obtained in Fig. 8*b* and *d* suggesting that the pairwise relationship does not have a significant effect over the synthetic centroid derivation.

A further analysis was done by comparing the results obtained using Lasso-based network (since they have the best performance) with those associated to two most used scenario reduction approaches, SCENRED and OSCAR. Note that for SCENRED and

OSCAR, the reduced set size must be defined a priory, so, the same sizes as the ones obtained in Lasso approach were used. The optimal results for SCENRED and OSCAR are also displayed in Table 2 and the deviation from the full-space are in Fig. 8. SCENRED and OSCAR performances show similar behaviour in which, the larger the size of the reduced set of scenarios, the lower approximation error.

These results suggest that SCANCODE (using Lasso-based networks) is a useful alternative to reduce the superset of scenarios closing the gaps between EP, PP and FS at least as much as the current scenario reduction approaches.

3.3 Design of a Hybrid Biofuel Supply Chain

The second case study addresses the optimal design of a multi-period optimisation framework for a hybrid bio-ethanol supply chain network in the UK, using wheat and wheat straw as first and second-generation feedstocks respectively. Three periods have been considered (e.g. 2012-2014, 2015-2017, 2018-2020), while the UK territory is equally divided into 37 regions with a total of 6 biofuel demand centres (See Fig. 9). Particularly, four uncertain parameters have been considered (biomass supply, biomass imports, biofuel sales and import prices). A total of 500 scenarios have been generated following a uniform distribution for these parameters considering a range of $\pm 50\%$ with respect to their average values for biomass supply, biomass imports and biofuel sales while for import prices an increment within the range of 10 to 50% was considered. For more details regarding the mathematical formulation the readers care referred to the supplementary material file while the main problem parameters can be found in (Akgul et al., 2011b,a).

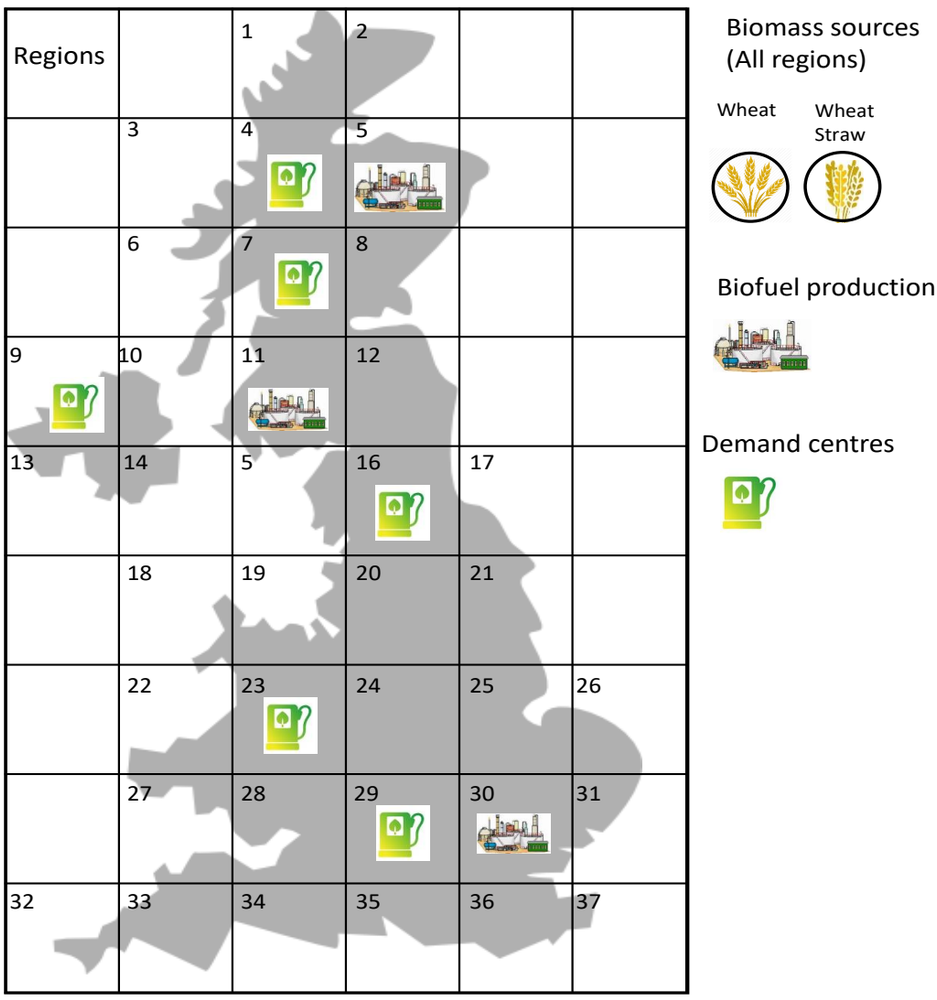

Figure 9: Superstructure of the Biofuel Supply Chain across UK

Following the same validation approach two different adjacency matrices have been generated using five different Th/λ values (*Th=0.96, Th=0.97, Th=0.98, Th=0.99* and *Th=0.995* for the Pearson correlation and λ=0.015, λ=0.051, λ=0.088, λ=0.115 and λ=0.197 for Lasso). The associated networks were divided into 10, 14, 18, 38 and 72 communities for Pearson-based networks and 12, 18, 32, 63 and 187 of them for Lasso-based networks. Note that in this case, the Pearson correlation achieves significantly larger reductions if compared to Lasso-based networks. Nevertheless, the associated networks at similar cluster numbers confirm that Lasso-based networks achieve better network construction (i.e. less dispersed points/clusters) (see Fig. 10). Additionally, these results prove that Lasso-based networks are better suited at building networks regardless of the number of uncertain parameters as well as the distribution used to generate the original dataset.

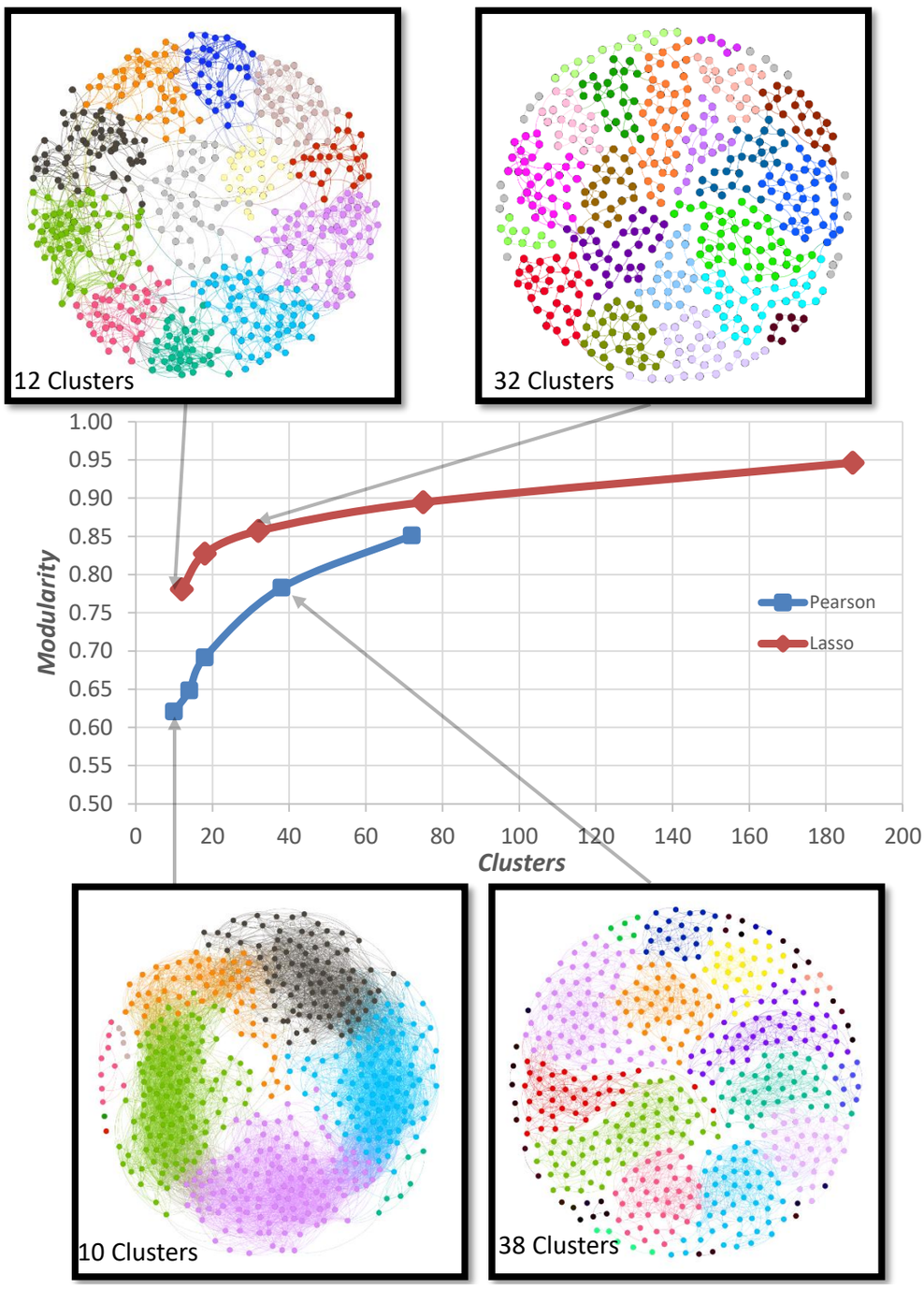

Figure 10: Relation between modularity value and clusters size for Biofuel Supply Chain problem

Once again, closeness metric and synthetic centroid approaches are explored to identify/derive the cluster centroids per cluster.

3.3.1 Case 2: Process optimisation results

The results of the MILP problem solved using the reduced set of scenarios obtained by the examined methods are summarised in Table 3.

Table 3: Results of the Biofuel Supply Chain optimisation of MILP problem

	Clusters	***Modularity***	***ExpProfit[a] (EP)***	***PostProcess[a] (PP)***	***CPU seconds***
Full-space (FS)	500		1.3690		141,506.81
			Pearson-based network		
Th			*Closeness centroid*		
0.960	10	0.6206	0.1781	1.0935	6.172
0.970	14	0.6482	1.0533	1.3687	15.638
0.980	18	0.6917	0.5973	1.3417	12.594
0.990	38	0.7827	1.0918	1.3690	67.875
0.995	72	0.8513	1.4599	1.3687	1,162.03
Th			*Synthetic Centroid-Pearson*		
0.960	10	0.6206	1.4944	1.3611	2.328
0.970	14	0.6482	1.3820	1.3686	3.829
0.980	18	0.6917	1.5515	1.3611	8.703
0.990	38	0.7827	1.3828	1.3686	30.344
0.995	72	0.8513	1.4606	1.3686	1,003.12
			Lasso-based network		
λ			*Closeness centroid*		
0.015	12	0.7806	1.9197	1.3690	9.829
0.051	18	0.8274	1.7265	1.3690	7.563
0.088	32	0.8572	1.2181	1.3416	48.828
0.115	63	0.8946	1.1251	1.2431	753.656
0.197	187	0.9464	1.5288	1.3690	3,361.28
λ			*Synthetic Centroid-Lasso*		
0.015	12	0.7806	1.4244	1.3686	8.562
0.051	18	0.8274	1.3025	1.3677	5.824
0.088	32	0.8572	1.2974	1.3416	43.254
0.115	63	0.8900	1.1298	1.2431	653.624
0.197	187	0.9464	1.3951	1.3690	1,395.26
			SCENRED		
–	12	–	1.2975	1.3399	60.531
–	18	–	1.4016	1.3399	48.609
–	32	–	1.3491	1.3416	271.891
–	63	–	1.3455	1.3399	1,849.67
–	187	–	1.3283	1.3399	9,971.41
			OSCAR		
–	12	–	1.1755	1.3416	49.456
–	18	–	1.3465	1.3656	42.015
–	32	–	1.4478	1.3686	311.109
–	63	–	1.7129	1.3690	1,938.81
–	187	–	1.5911	1.3690	5,498.28

a (€x10^9)

Similarly to the previous example, the deviation from the *FS* performance was calculated (see Fig. 11). These results demonstrate that, in contrast to the results in the first case, for this particular problem the synthetic centroid performs better at reducing the gaps between FS and EP. This might be due to the different probability distributions used in the generation of the original set of scenarios and highlights the flexibility of this approach as well as its capability to be adapted based on the problem needs. Remarkably, for this case, the computational savings achieved by SCANCODE reach one order of magnitude less compared to OSCAR and SCENRED. Such a computational performance justifies the use of this approach over their current counterparts.

Numerical results also demonstrate that both, SCANCODE (synthetic Lasso-based network) and SCENRED accurately approximate the *EP* regardless of the reduction

level. OSCAR on the other hand presents considerable large gaps for most of the instances. Disregarding such a close gap between *EP* and *FS*, a *PP* vs *FS* analysis was performed for all the methods to validate the quality of the reduced set of scenarios over the whole uncertainty space. The gaps between the *EP*, *PP* and *FS* are visually illustrated in Fig. 11.

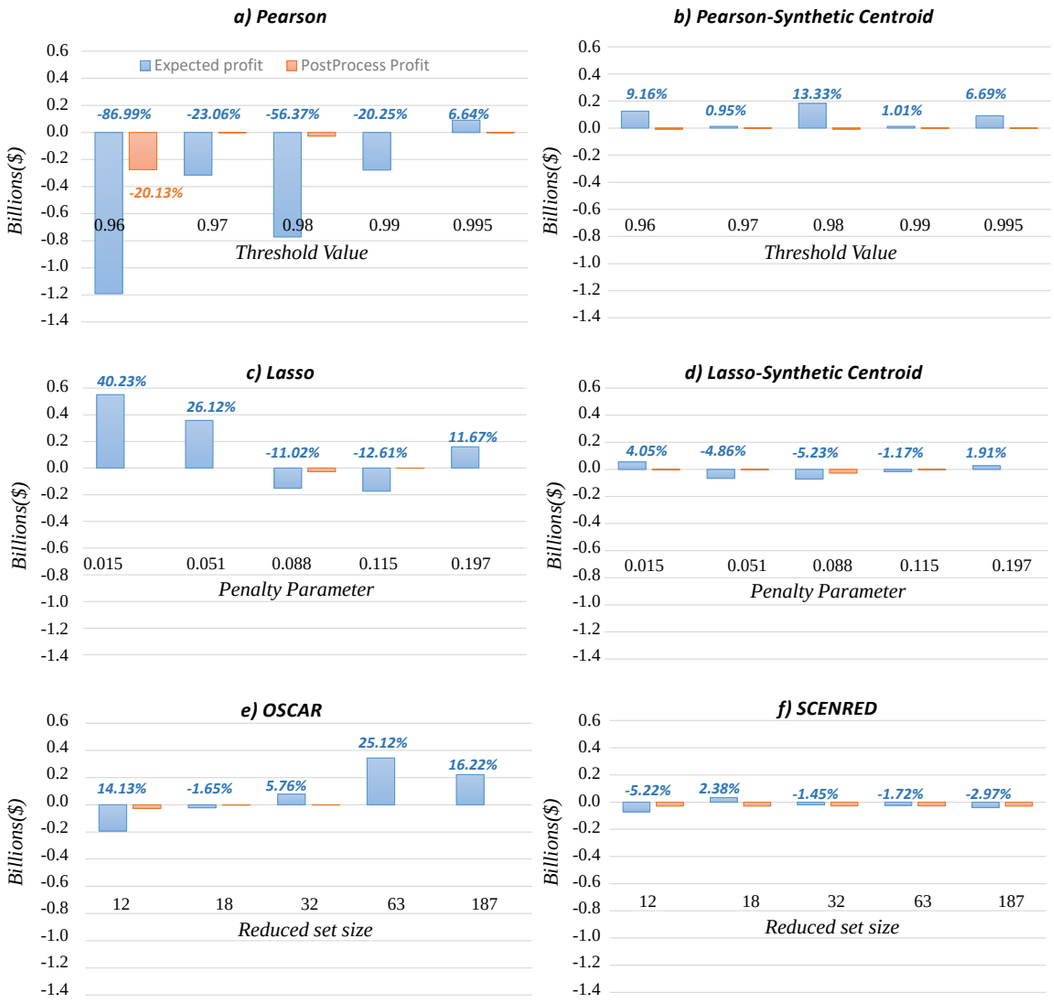

Figure 11: Deviation of the optimal solutions for 5 different levels of reduction for the Biofuel Supply Chain.

A considerably close gap between the *PP* and *FS* profit was obtained for all the approaches, with the exception of a single threshold value for Pearson ($Th = 0.96$). Disregarding this isolated event, both, SCANCODE and OSCAR, reduce the gap between *FS* and *PP* better than SCENRED for which a minimum gap of 2% is achieved. Nevertheless, comparing *EP* with *FS*, SCANCODE (using synthetic centroid for Lasso-based networks) achieves the overall better results, closely followed by SCENRED which are the only cases in which the gap can be reduced below 5%.

Note that in general, SCENRED and SCANCODE have proved to be reliable approaches since they provide small gaps between *EP*, *PP* and *FS*. Nevertheless, in order to stress even more the capabilities of these approaches and produce a more detailed conclusion, the cumulative distribution plots have been analysed to evaluate the process behaviour of the approaches. Fig. 12 demonstrates that SCANCODE is more accurate capturing the process behaviour which is of great relevance, especially when

detailed analysis is needed/desired (e.g. while calculating the financial risk or any other distributional-based metric).

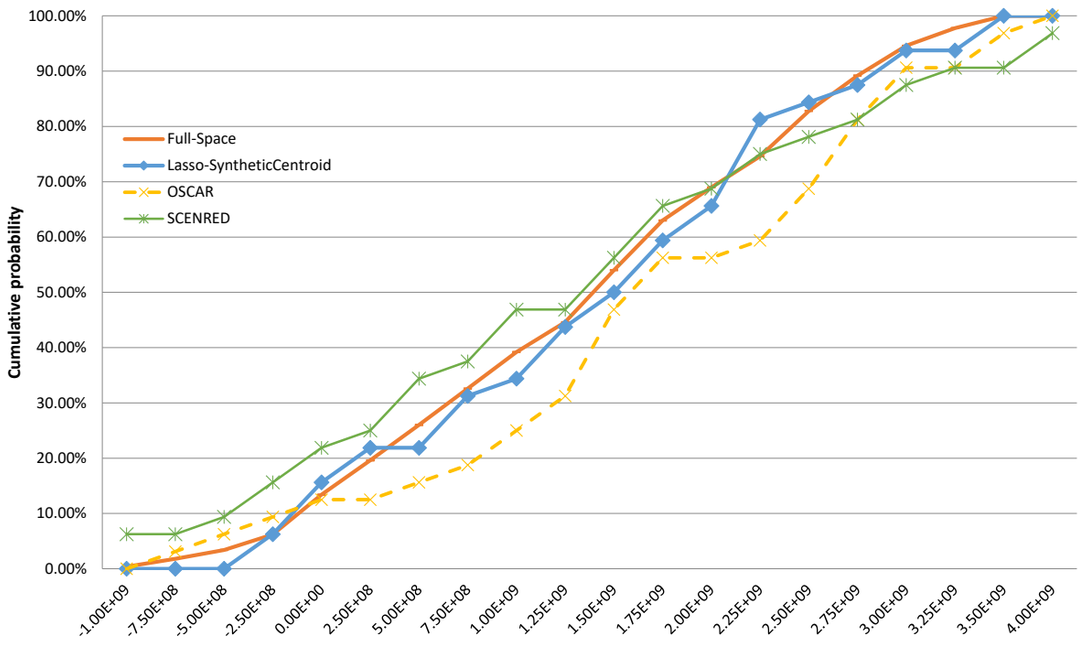

Figure 12: Cumulative distribution for the 32 reduced set of scenarios

The above results, confirm that the proposed approach is as good as OSCAR and SCENRED (in some cases better) to represent a set of uncertainty conditions generated following uniform probabilistic distributions using a reduced set of elements.

3.4 Design and management of a Water network

The final case study aims for the maximal economic benefit of a water system in an urban area of Mexico through its optimal design and planning. 12 natural water sources were considered with a maximum water usage of 80% of their capacity. The problem aims for the optimal distribution of water that satisfies the demands of different users (domestic, agricultural, and industrial) as illustrated in Fig. 13. To describe the uncertain demand for the five-year time horizon a total of 5,000 scenarios with 180 uncertain parameters were generated following a normal distribution and a standard deviation of 30%. The associated MILP problem taken from (Medina-González et al., 2018b) was summarised in the supplementary material file.

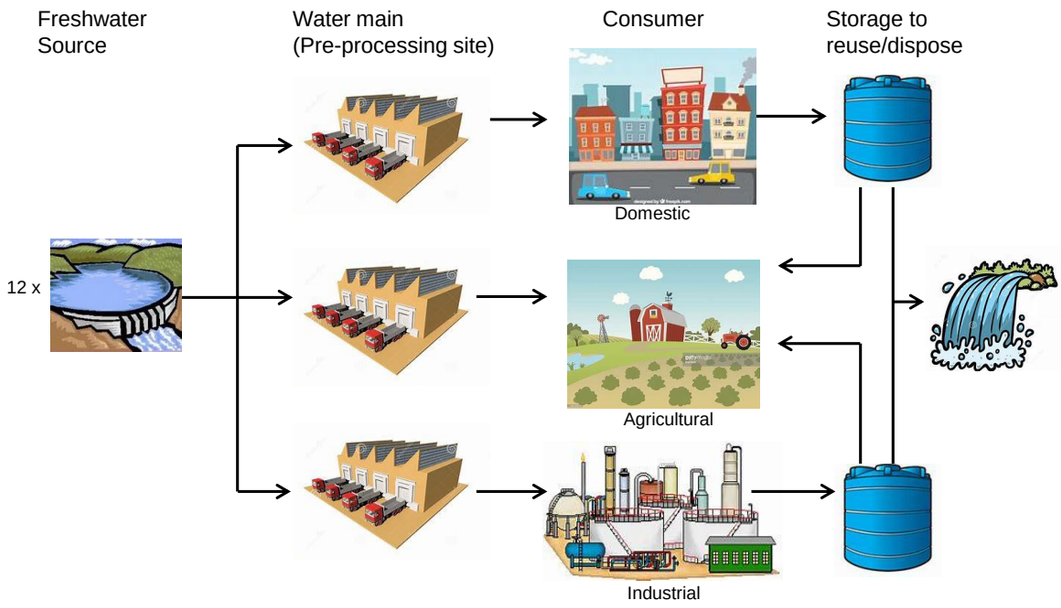

Figure 13: Water network illustration

For this example, four Th/λ values were used (specifically *Th=0.86, Th=0.865, Th=0.87 and Th=0.875* for Pearson correlation and λ=0.050, λ=0.070, λ=0.080 and λ=0.085 for Lasso regression). The associated networks consist of 9, 15, 30 and 80 communities for Pearson-based networks and 7, 18, 24 and 82 of them for Lasso-based ones. Even though both approaches achieve similar reductions, Pearson's fails to produce meaningful networks, since the Newman's network quality rule is not satisfied (i.e. modularity values below 0.3 are obtained) as presented in Fig. 14. For the case of Lasso regression, any cluster size ≥ 24 leads to strongly connected networks. These results prove that Lasso-regression remains efficient constructing a meaningful network independently of the number of nodes in the original dataset.

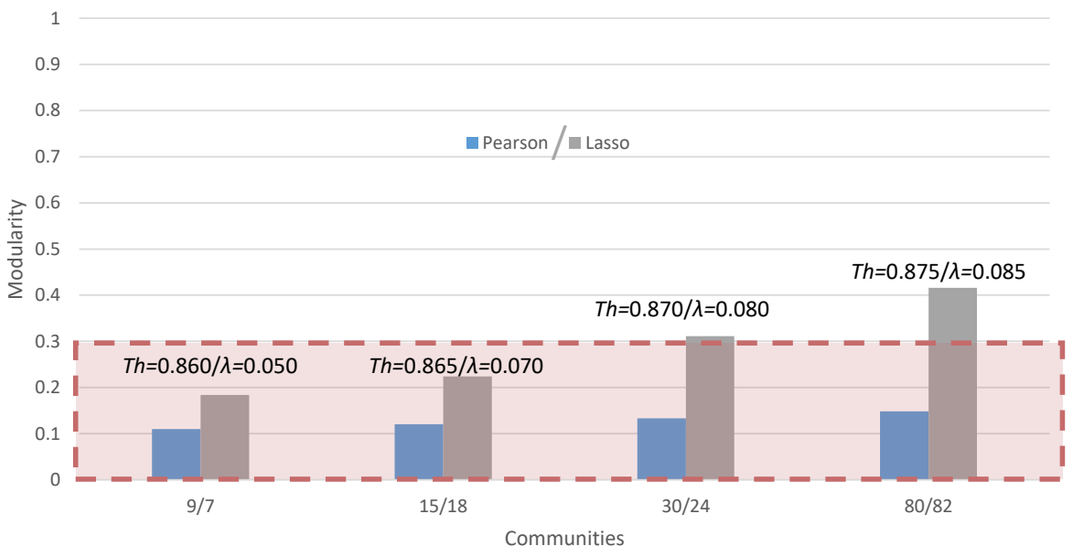

Figure 14: Relation between modularity value and clusters size

3.4.1 Case 3: Process optimisation results

The results of the MILP problem solved using the reduced set of scenarios for SCANCODE, OSCAR and SCENRED are summarised in Table 4. Note that due to the high computational effort required to solve the *FS*, (a solution below 5% cannot be obtained at a reasonable time $\leq$ 100,000 seconds), the solution was obtained by calculating the average of the 5,000 deterministic solutions. Such an approximation was used as a crude relaxation of the complex problem only to get an idea of the expected profit of the system solved rigorously using two-stage stochastic programming and illustrate the advantage of SCANCODE over the other approaches.

Table 4: Profit deviation between different approaches for the Water Distribution network.

	Clusters	***Modularity***	***ExpProfit (EP)***	***PostProcess (PP)***	***CPU seconds***
Full-space (FS)	5,000		250,618,890		552,453
			Pearson-based network		
Th			*Closeness centroid*		
0.860	9	0.110	251,067,610	250,555,100	153
0.865	15	0.120	250,189,475	250,634,660	464
0.870	30	0.133	250,619,790	250,572,671	37,960
0.875	80	0.148	244,750,841	250,572,671	100,000
Th			*Synthetic centroid-Pearson*		
0.860	9	0.110	250,650,380	250,535,504	126
0.865	15	0.120	250,580,982	250,472,932	239
0.870	30	0.133	250,564,950	250,470,208	667
0.875	80	0.148	246,971,433	250,470,208	100,000
			Lasso-based network		
λ			*Closeness centroid*		
0.050	7	0.184	246,607,441	250,539,609	317
0.070	18	0.224	249,257,179	250,627,491	760
0.080	24	0.311	249,614,522	250,622,903	14,873
0.085	82	0.416	248,634,987	250,622,903	100,000
λ			*Synthetic centroid-Lasso*		
0.050	7	0.184	250,729,371	250,570,614	64
0.070	18	0.224	250,691,310	250,566,561	565
0.080	24	0.311	250,640,960	250,534,373	20,540
0.085	82	0.416	249,972,325	250,534,373	100,000
			SCENRED		
–	7	–	248,506,555	250,462,308	104
–	18	–	249,824,661	250,640,348	564
–	24	–	250,397,998	250,604,357	25,136
–	82	–	250,243,940	250,604,357	100,000
			OSCAR		
–	7	–	246,465,935	250,532,353	79
–	18	–	246,628,089	250,566,561	262
–	24	–	250,019,790	250,589,201	27,498
–	82	–	249,243,940	250,589,201	100,000

Similarly to the second case study, the synthetic centroid approximates better the *EP* and *PP* compared to the *FS*. These results support the idea of using synthetic centroid for medium/large size datasets regardless of the distributions used to generate it. Also, in Fig. 15 the advantage of using Lasso with synthetic centroid over SCENRED and OSCAR is clearly demonstrated since the differences between *EP*, *PP* and *FS* are

minimal.

Notice that SCANCODE displays an overall reduction in the computational time compared to SCENRED and OSCAR, which emphasises its advantages while facing larger problems. Remarkably, the computational savings were not as significant as in the second case, which could be attributed to the use of 2% as the optimality gap. Thus, a larger difference can be expected by using an optimality gap of 0%.

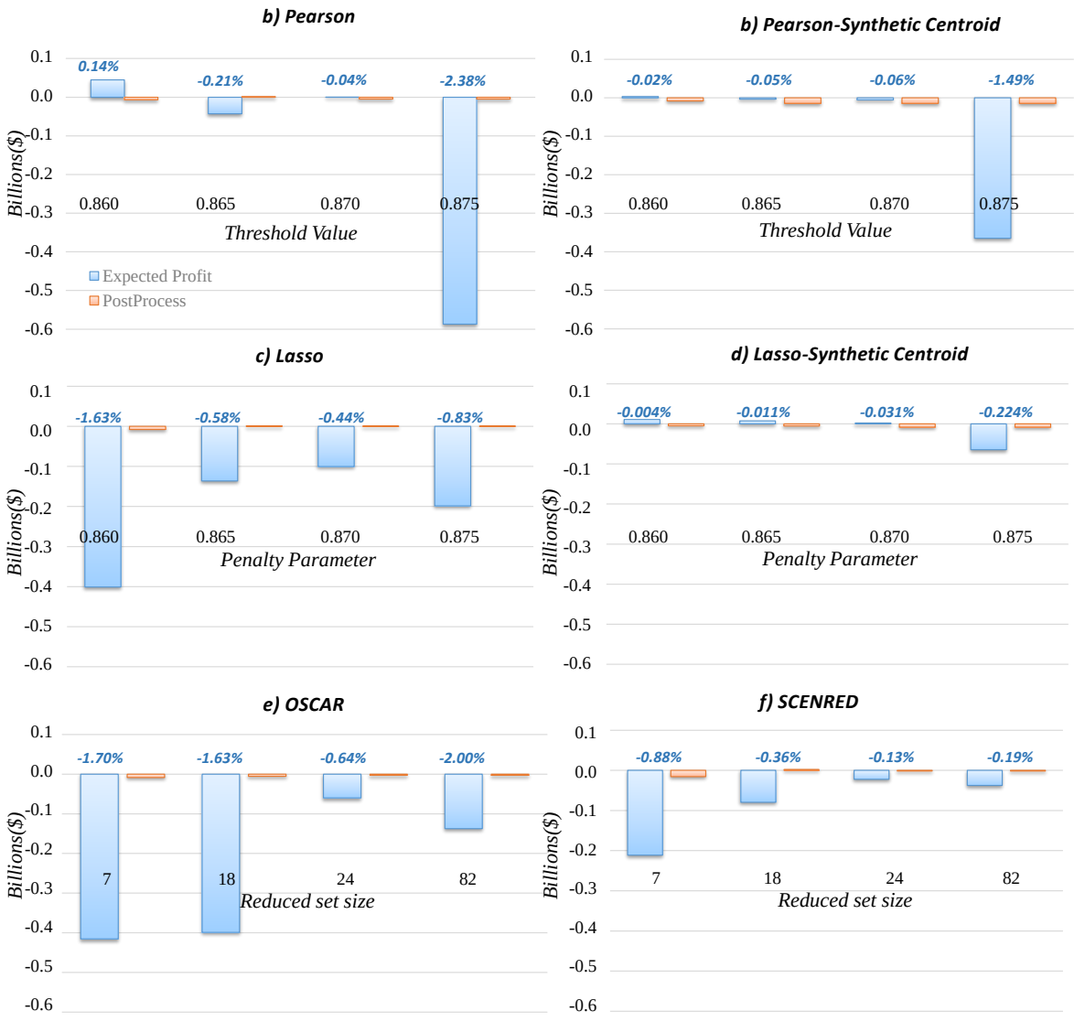

Figure 15: Deviation of the optimal solutions of 4 different levels of reduction for the Water distribution network.

4 Concluding remarks

This paper proposes SCANCODE as a novel scenario reduction/aggregation framework through the integration of graph theory and community detection methods. The proposed method is a user-induced approach in which the user defines a Th/λ value that directly affects the network density. Later, using that network, the algorithm determines the subset of scenarios that better represents the input-data. A parameter definition rule was proposed to reduce the subjectivity in the decision-maker intervention. SCANCODE's robustness and flexibility was proved by allowing the use of different techniques/methods within the same framework to address different types of problems. Numerical results prove that the proposed strategy identifies a reduced and representa-

tive set of scenarios, while its reliability was demonstrated by comparing those results with the ones obtained using SCENRED and OSCAR. In general, SCANCODE performs equally good (and in some cases better) at approximating the expected profit and the associated post-process performance. Remarkably, SCANCODE allows an evaluation of the subset of scenarios prior the optimisation, which represents a desirable property towards application. Additionally, it has been proved that for problems with large original scenarios set or low dimensionality, SCANCODE provides accurate and quick results which represents a step-forward for scenario reduction techniques. These results justify the consideration of SCANCODE to be used for large-scale networks considering many uncertain scenarios (of several orders of magnitude).

In general, Lasso-based networks perform better than Pearson-based ones for all the cases. Within this networks, the synthetic centroid leads to a more accurate representation of the process performances for a relatively large set of scenarios (> 500) while the definition of a node as cluster centre is efficient for small datasets. Further research is needed to define a single centroid definition regfardless of the problem peculiarities.

Even though SCANCODE proves to be a useful scenario reduction method, there is a need for an integrated tool within some mathematical modeling and optimisation software to compete with SCENRED at a practical level. Additionally, some opportunity areas have been identified that will improve SCANCODE performance and ease its applicability. Particularly, higher modularity values and further reduction of both, isolated elements considered as scenarios/clusters and expected profit gaps are open challenges in this approach. Moreover, further research is needed to have full control over the number of resulting clusters so as to instead of defining the threshold value be able to directly identify the optimal number of scenarios in the reduced set to accurately represent the uncertainty space.

5 Nomenclature

Abbreviations	
RedOpt	Scenario reduction algorithm proposed by Silvente et al. (2019).
SCENRED	Scenario reduction tool available in GAMS.
OSCAR	Scenario reduction algorithm proposed by Li and Floudas (2016).
CR	Cassava Rhizome.
FS	Full-space solution
EP	Expected profit
PP	Postprocess profit
MILP	Mixed integer linear programming
Indices	
s	Scenario
i	Node/vertex
j	Node/vertex
m	Communities
t	Time period
Parameters	
$A_{i,j}$	Adjacency matrix
d	Geodesic distance
l_i	Closeness centrality metric for each node
Th	Threshold value for Pearson's correlation
λ	Penalty parameter for Lasso regression
Tr	Trace matrix
Q	Modularity value
pr_s	Probability of occurrence for scenario s
$e_{m,m'}$	Matrix that represents the fraction of the edges connecting elements in community m and elements in community m'
a_m	Fraction of edges connecting vertices within community m

Acknowledgement

The authors would like to thank the financial support received from the UK Engineering and Physical Sciences Research Council (under the project EP/M028240/1) as well as the UK Leverhulme Trust under grant number RPG-2015-240.

Supplementary material

The supplementary material associated to this article can be found in the online version.

Appendix A Background theory

A.1 Graph theory

A graph or network is a collection of points that are connected between them. The points or vertices are known as *nodes* and the links/lines as *edges*. Networks have been used to describe a variety of systems and fields such as biological, social and computer networks (Fortunato, 2010).

A network structure is defined by the adjacency matrix. This matrix contains elements representing the connectivity of nodes in a network. If {i,j} are both used to represent nodes and A is the adjacency matrix, then:

$$A_{i,j} = \begin{cases} 1 & \text{if there is an edge between nodes } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

In the simple case of an unweighted and undirected network, this matrix is symmetric and only contains elements with values {0,1}. In the case of weighted networks, the matrix contains values equal to the weights of the corresponding edges. Directed networks contain the additional information of directionality, with edges having a specific direction from one node to the other, which typically results in a non-symmetric matrix (Newman, 2018).

Since complex systems can be visualised as networks, it is natural that some nodes will be different from others based on some network properties. One such property is the centrality of a node, which is a metric that quantifies how central a node is compared to others. Some common metrics from the literature are the following (Newman, 2018):

Degree centrality

The most central node is the one that has the maximum degree, meaning the node which has the highest number of edges connected to it. In directed networks vertices have both an in-degree and an out-degree, and both may be useful as measures of centrality in the appropriate circumstances.

Closeness centrality

This metric measures the mean distance of a node to other nodes. If d is the geodesic path between two nodes i and j, meaning the shortest path between these two nodes, then the mean distance in a network with n nodes is defined as:

$$l_i = \frac{1}{n} \sum_j d_{ij}$$

Because this definition gives ‘low‘ distance values for central nodes, it is very common to use the inverse of the mean distance.

These are two common and frequently used centrality metrics, but there are also more definitions in the literature such as eigenvector centrality, betweenness centrality and

others.

A.2 Community detection

In real networks, it is very common to identify areas where the vertices are naturally clustered into local areas with high concentrations of edges within them and low between other groups. This feature of real networks is called *community structure* or *community detection* (Girvan and Newman, 2002). This procedure of identifying 'meaningful' clusters in real networks, can have multiple applications such as the world wide web, where clusters can reveal relationships of websites, or in social networks where each cluster could correspond to different local communities and others.

A number of computational approaches have been proposed over the years to tackle this problem, with common methods including graph partitioning and hierarchical clustering. Further investigation of networks led to the introduction of a new metric called *modularity* (Newman and Girvan, 2004). This metric measures the fraction of edges in a network that connects nodes of the same type (i.e., within-community edges) minus the expected value of the same quantity in a network with the same community divisions, but random connections between the nodes.

A.3 Modularity metric

Consider a particular division of a network into m communities. Let us define an mxm symmetric matrix e whose element $e_{mm'}$ is the fraction of all edges in the network that link vertices in community m to vertices in community m'. The trace of this matrix Tr $e = \sum_m e_{mm}$ gives the fraction of edges in the network that connects vertices in the same community, and clearly, a good division into communities should have a high value of this trace.

Furthermore, the row (or column) sums $a_m = \sum_{m'} e_{mm'}$, which represent the fraction of edges that connect the vertices in the community m. In a network in which edges fall between vertices without regard for the communities they belong to, the modularity value can be express as:

$$Q = \sum_m (e_{mm'} - a_m^2) = Tre - \left\| e^2 \right\| \tag{A.3.1}$$

where $\|e^2\|$ indicates the sum of the elements of the matrix e. This quantity measures the fraction of the edges in the network that connect vertices within communities minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices. Logically, if the number of within-community edges is no better than the randomly generated connections, we assume a poorly or disconnected network (Q=0), while values approaching the maximum value (i.e. Q=1) indicate networks with strong community structure. In practice, Q values ranging between 0.3 to 0.7 are considered strong structure communities since higher values are rare (Newman and Girvan, 2004).

A.4 Pearson correlation coefficient

Pearson's correlation coefficient describes the association between variables. For any two variables X and Y, the coefficient measures the linear correlation between those two variables. The formula is given below (Hastie et al., 2008):

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{A.4.1}$$

Where:

$cov(X,Y)$	covariance of variables X and Y
σ_X, σ_Y	standard deviations of variables X and Y
$corr(X,Y)$	the correlation value

This coefficient takes values from -1 to 1, and they represent the percentage of linear correlation.

A.5 Lasso Regression

Lasso is a regression analysis method that uses the L1 norm to penalise the regression coefficients to perform variable selection and regularisation in order to improve the predictive accuracy of a model. If a dataset of $i = 1, 2, ..., N$ observations has $j = 1, 2, ..., p$ variables and an output y_i, then Lasso can be written in the *Lagrangian* form as (Hastie et al., 2008):

$$argmin \left\{ \frac{1}{2} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

where β_0, β_j are the regression coefficients and λ is the penalty parameter that is user-defined and controls the size of the penalty.

A.5.1 Penalty initialisation

The calculation of an appropriate value for the λ parameter requires a sensitivity analysis. In this work, the selected approach is k-*fold* cross-validation. For each value of λ, k-*fold* validation splits the dataset into k partitions and uses 1 partition for validating purposes and the rest k-1 for training the model. The process is repeated until all k partitions have been used for validation. The error between observed values and model predictions are captured and the overall average is reported at the end.

The entire implementation has been performed in the R programming language, using the `glmnet` package. This package enables the user to perform k-*fold* validation while automatically choosing a sequence of λ values. The outcome is a vector of average error values for all the different values of λ.

Considering the `glmnet` capabilities and knowing that in this work the appropriate λ value is unknown to the user, an iterative heuristic approach is used to propose a range

of values. Essentially, this approach employs the mentioned k-*fold* validation scheme repeated for every combination of scenarios (those combinations are described in section 2.2.2). The outcome is an optimal λ value that corresponds to the minimum average error for each scenario. So, if λ_values are the vector of all the optimal lambda values, then the heuristic provides the following range:

$$average(\lambda\_values) \pm \sigma(\lambda\_values)$$

where σ is the standard deviation.

Selecting a value for λ that lies within that range is a good starting point for constructing the adjacency matrix and create a network.

References

Ahmed, S. and Sahinidis, N. (2003). An approximation scheme for stochastic integer programs arising in capacity expansion. *Operations Research*, 51(3):461–471.

Akgul, O., Shah, N., and Papageorgiou, L. G. (2011a). An MILP model for the strategic design of the UK bioethanol supply chain. *21st European Symposium on Computer Aided Process Engineering*, 29:1799–1803.

Akgul, O., Zamboni, A., Bezzo, F., Shah, N., and Papageorgiou, L. G. (2011b). Optimization-based approaches for bioethanol supply chains. *Industrial & Engineering Chemistry Research*, 50:4927–4938.

Bhagat, A., Kshirsagar, N., Khodke, P., Dongre, K., and Ali, S. (2016). Penalty parameter selection for hierarchical data stream clustering. *Procedia Computer Science*, 79:24–31.

Birge, J. and Louveaux, C. (1997). *Introduction to Stochastic Programming*. Springler-Verlag, New York.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008.

Calafiore, G. and Campi, M. (2005). Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102:25–46.

Celebi, M., Kingravi, H., and vela, P. (2013). A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210.

Chen, Z., Peng, S., and Liu, J. (2018). Data-driven robust chance constrained problems: A mixture model approach. *Journal of Optimization Theory and Applications*, 179:1065–1085.

Chen, Z. and Yan, Z. (2018a). Practical arbitrage-free scenario tree reduction methods and their applications in financial optimization. *Wiley*, 34:175–195.

Chen, Z. and Yan, Z. (2018b). Scenario tree reduction methods through clustering nodes. *Computers & Chemical Engineering*, 109:96–111.

Chu, Y. F. and You, F. (2013). Integration of scheduling and dynamic optimization of batch processes under uncertainty: Two-stage stochastic programming approach and enhanced generalized benders decomposition algorithm. *Industrial & Engineering Chemistry Research*, 52:16851–16869.

Dupacova, J. and Gröwe-Kuska, and Römisch, W. (2003). Scenario reduction in stochastic programming:an approach using probability metrics. *Mathematical Programming SERIES A*, 95:493–511.

Esfahani, P. and Kuhn, D. (2018). Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171:115–166.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486:75–174.

GAMS documentation (2019). SCENRED: Scenario reduction algorithms. *Available at* https://www.gams.com/latest/docs/tools/scenred/index.html.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99:7821–7826.

Growe-Kuka, N., Heitsch, H., and Romisch, W. (2003). Scenario reduction and scenario tree construction for power management problems. volume 3, pages 1–7. IEEE.

Hastie, T., Tibshirani, R., and Friedman, J. (2008). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2 edition.

Heitsch, H. and Römisch, W. (2003). Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, 24:187–206.

Heitsch, H. and Römisch, W. (2007). A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, 35:731–738.

Heitsch, H. and Römisch, W. (2009). Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118:371–406.

Henrion, R., Küchler, C., and Römisch, W. (2008). Discrepancy distances and scenario reduction in two-stage stochastic mixed-integer programming. *Journal of Industrial & Management Optimization*, 4:363–384.

Henrion, R., Küchler, C., and Römisch, W. (2009). Scenario reduction in stochastic programming with respect to discrepancy distances. *Computational Optimization and Applications*, 43:67–93.

Javed, M., Younis, M., Latif, S., Qadir, J., and Baig, A. (2018). Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 108:87–111.

Karuppiah, R., Martin, M., and Grossmann, I. E. (2010). A simple heuristic for reducing the number of scenarios in two-stage stochastic programming. *Computers & Chemical Engineering*, 34:1246–1255.

Kusiak, A. (2006). Data mining: manufacturing and service applications. *International Journal of Production Research*, 44:4175–4191.

Li, Z. and Floudas, C. A. (2016). Optimal scenario reduction framework based on distance of uncertainty distribution and output performance: II. sequential reduction. *Computers & Chemical Engineering*, 84:599–610.

Li, Z. and Ierapetritou, M. (2012). Capacity expansion planning through augmented lagrangian optimization and scenario decomposition. *AIChE Journal*, 58:871–883.

Li, Z. and Li, Z. (2016). Linear programming-based scenario reduction using transportation distance. *Computers & Chemical Engineering*, 88:50–58.

Lima, C., Relvas, S., and Barbosa-Póvoa, A. P. (2018). Stochastic programming approach for the optimal tactical planning of the downstream oil supply chain. *Computers & Operations Research*, 108:314–336.

Medina-González, S., Espuña, A., and Puigjaner, L. (2018a). An efficient uncertainty representation for the design of sustainable energy generation systems. *Chemical Engineering Research and Design*, 131:144–159.

Medina-González, S., Graells, M., Guillén-Gosálbez, G., Espuña, A., and Puigjaner, L. (2017). Systematic approach for the design of sustainable supply chains under quality uncertainty. *Energy Conversion and Management*, 149:722–737.

Medina-González, S., Rojas-Torres, M., Ponce-Ortega, J., Espuña, A., and Guillén-Gosálbez, G. (2018b). Use of nonlinear membership functions and the water stress index for the environmentally conscious management of urban water systems: Application to the city of morelia. *ACS Sustainable Chemistry & Engineering*, 6:7752–7760.

Monostori, L. and Viharos, Z. J. (2001). Hybrid, ai- and simulation-supported optimisation of process chains and production plants. *CIRP Annals*, 50:353–356.

Newman, M. E. (2003). Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:1–5.

Newman, M. E. (2018). *Networks*. Oxford university press.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, 69:026113.

Ning, C. and You, F. (2018a). Adaptive robust optimization with minimax regret criterion: Multiobjective optimization framework and computational algorithm for planning and scheduling under uncertainty. *Computers & Chemical Engineering*, 108:425–447.

Ning, C. and You, F. (2018b). Data-driven stochastic robust optimization: General computational framework and algorithm leveraging machine learning for optimization under uncertainty in the big data era. *Computers & Chemical Engineering*, 111:115–133.

Ning, C. and You, F. (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering*, 125:434–448.

Römisch, W. (2009). Scenario reduction techniques in stochastic programming. In *Stochastic Algorithms: Foundations and Applications*, volume 5, pages 1–14.

Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers & Chemical Engineering*, 28:971–983.

Shang, C. and You, F. (2018). Distributionally robust optimization for planning and scheduling under uncertainty. *Computers & Chemical Engineering*, 110:53–68.

Silvente, J., Papageorgiou, L. G., and Dua, V. (2019). Scenario tree reduction for optimisation under uncertainty using sensitivity analysis. *Computers & Chemical Engineering*, 125:449–459.

Xu, B., Zhong, P.-A., Zambon, R. C., Zhao, Y., and Yeh, W. (2015). Scenario tree reduction in stochastic programming with recourse for hydropower operations. *Water Resources Research*, 51:6359–6380.

Xu, G., Tsoka, S., and Papageorgiou, L. G. (2007). Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B*, 60:231–239.

You, K., Tempo, R., and Xie, P. (2018). Distributed algorithms for robust convex optimization via the scenario approach. *IEEE Trans. Autom. Control*, 64:1–15.

Zeballos, Luis, J., Méndez, Carlos, A., and Barbosa-Póvoa, A. P. (2018). Integrating decisions of product and closed-loop supply chain design under uncertain return flows. *Computers & Operations Research*, 112:211–238.