CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Decision-Based Scenario Clustering for Decision-Making under Uncertainty

Michael Hewitt
Janosch Ortmann
Walter Rei

October 2018

CIRRELT-2018-39

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Decision-Based Scenario Clustering for Decision-Making under Uncertainty

Mike Hewitt[1], Janosch Ortmann[2,*], Walter Rei[2,3]

[1] Department of Information Systems and Supply Chain Management, Quinlan School of Business, Loyola University, 1 E. Pearson, Suite 204, Chicago, IL 60611, USA

[2] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

[3] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

Abstract. In order to make sense of future uncertainty, managers have long resorted to creating scenarios that are then used to evaluate how uncertainty affects decision-making. The large number of scenarios required to faithfully represent several sources of uncertainty leads to major challenges in using the scenarios in a decision-support context. Moreover, the complexity induced by the large number of scenarios can stop decision makers from reasoning about the interplay between the uncertainty modelled by the data and the decision-making processes. In order to meet this challenge, we propose a new approach to group scenarios based on the decisions associated to them. We introduce a graph structure on the scenarios based on the opportunity cost of predicting the wrong scenario. This allows us to apply graph clustering methods and to obtain groups of scenarios with mutually acceptable decisions. In order to test our approach, we apply it in the context of stochastic optimisation, specifically as a means to derive both lower and upper bounds for stochastic network design models and fleet planning problems under uncertainty. Our numerical results indicate that our approach is particularly effective to derive high-quality bounds when dealing with complex problems under time constraints.

Keywords: Stochastic optimisation, graph clustering, fleet planning, stochastic network design.

Acknowledgments. This research was partially funded by the Natural Sciences and Engineering Council of Canada (NSERC) via the Discovery Grant programme (WR) and by Concordia University through a Horizon postdoctoral fellowship (JO). This support is gratefully acknowledged. We are also grateful to Khedidja Seridi for her help in implementing the necessary code for the biweekly fleet planning problem.

---

* Corresponding author: ortmann.janosch@uqam.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2018

1. Introduction

Decision-makers are continuously solving problems in the presence of different sources and varying levels of uncertainty that affect the information parameters defining them, e.g., Pownuk and Kreinovich (2018) and King and Wallace (2012). The need to develop decision support methodologies that explicitly account for such uncertainty is undeniable. Numerous studies have shown how uncertainty can significantly affect decision-making processes in a variety of domains. When planning and managing inventories in supply chains, the bullwhip effect (e.g., see Metters (1997) and Ouyang and Li (2010)), which stems from the demand uncertainty of the end users, and its overall consequences (e.g., operational inefficiencies and excessive stocks throughout the chain) force organizations to implement specific planning strategies (e.g., that mainly rely on having better cooperation, coordination and communication between the actors of the chain, see Mackelpranga and Malhotra (2015)) to mitigate this problem. In the context of investment planning, the effects that higher levels of uncertainty have on reducing the responsiveness of organizations to adjust their investment strategies following demand shocks is also a well studied phenomenon, see Bloom et al. (2007).

Moreover, solutions to optimization models that are efficient in randomly varying environment are structurally different than solutions obtained by solving a deterministic optimization model, where perfect knowledge of the available information is assumed (Lium et al., 2009). In this context, scenario generation has been used as a cornerstone methodology to support decision-makers to both formulate how uncertain parameters vary and to explore how such uncertainty affects the decisions to be made.

In all generality, scenarios have first been used to approximate probability distributions that are applied to express stochastic parameters in optimization settings. Examples of such applications are numerous. They can be found in decision analysis methods, see Borgonovo et al. (2018), where event trees are created to assess the direct consequences of decisions when random events occur, but also as a means to evaluate how the notion of regret (or opportunity loss) affects how decisions are made when facing uncertainty, see Bell (1982). Using scenario generation to approximate probability distributions is also an integral part of how stochastic optimization models are both formulated and solved, see Birge and Louveaux (2011). In this case, representative scenarios are used to formulate recourse cost functions, which are then applied in stochastic optimization models to evaluate the projected costs of solutions (i.e., the future cost of decisions which are required to be made before the random events are observed). Various probabilistic methods have been proposed to generate such scenarios, e.g., Löhndorf (2016) and Høyland et al. (2003). Such methods have been successfully applied both statically to obtain solutions to complex stochastic optimization models, as in the case of stochastic network design models, see Crainic et al. (2011) and Rahmaniani et al. (2018); and dynamically to derive probabilistic bounds in the process of performing stochastic optimization, e.g., as in the cases of the sample average approximation method, see Kleywegt et al. (2002), and the stochastic decomposition strategy, see Higle and Sen (1991).

The second major use of scenarios has been to help perform planning processes in general managerial contexts, see Godet (2000). In this case, scenario generation (or scenario

analysis) refers to the tools used by managers, whose tasks involve solving planning problems at different decisional levels, to define a set of future informational outcomes that are then used to support them in these tasks. Such outcomes can define the informational contexts that are likely to be observed, but they can also represent specific perspectives, or intuitions, regarding possible futures that the managers may be interested to explicitly consider in the planning. As a matter of fact, to perform efficient long-term planning, an organization's foresight capabilities are often cited as one of the most important aspects, see Peter and Jarratt (2015).

The impact and relevance that information technologies have on organizational performance has been the subject of numerous scientific studies, see Melville et al. (2004) and Trieu (2017) for literature reviews on this general subject. As illustrated by Devaraj and Kohli (2003), one of the key factors that defines how information technologies influence organizational performance is their actual use. To apply scenario generation methods for decision-support, one of the main challenges resides in how to efficiently use the potentially high-volume of information (i.e., scenarios) that may be produced. On the one hand, instantiating a specific decision model with a large number of scenarios may render it numerically intractable to apply. For example, a stochastic optimization model that is formulated using a large set of scenarios may require a prohibitive amount of computational effort to solve. This is often the case when attempting to solve general stochastic integer programs, see Birge and Louveaux (2011). On the other hand, the complexity involved in assessing the managerial impact that a set of generated scenarios has on a specific decisional problem (e.g., through what-if analyses) may become insurmountable when the size of the set is very large. Actually, the various problems related to information overload for organizations, specifically as they may hinder decision processes, have become unavoidable, see Edmunds and Morris (2000). When applying scenario generation, one can actually reach a point where the amount of information to process simply becomes to great to distill meaningful insights.

In the present paper, we propose a general methodology that relies on graph clustering methods to identify structure in the scenario space associated with decision-making contexts under uncertainty. Specifically, for a given decision problem where the informational uncertainty is expressed through a set of generated scenarios (regardless of the strategy employed to generate them), our methodology can be used to identify groups of scenarios based on their proximity evaluated on a decisional basis. Our work makes the following contributions:

(1) We provide a novel provide to clustering scenarios on a decisional basis, using graph clustering methods
(2) We introduce new upper and lower bounds based on these clusters of scenarios
(3) We test this approach on two stochastic optimisation applications, namely stochastic network design and fleet planning.

The rest of the paper is divided as follows. We give a brief outline of the problem in Section 2. Our methodology is explained in Section 3, followed by our numerical results in Section 4. Section 5 concludes.

2. Problem statement

In its most general form, the kind of problem we attack in this paper can be stated as follows: given a probability measure P on a probability space $(\Omega, \mathcal{F})$ and a set $\mathcal{Y}$, we wish to find

$$\phi(P) = \inf_{y \in \mathcal{Y}} \Phi(y, P), \tag{2.1}$$

where Φ is a given function. This general formulation can be extremely complex, given that it attempts to define how a set of decisions to be made (represented by y) is to be evaluated subject to uncertainty (represented by P). In order to define a criterion on which decisions are to be evaluated, the mean or expectation value is often used. For this reason, a natural choice for the function Φ is given by

$$\Phi(y, P) = E_P(g(y); \omega) \tag{2.2}$$

for a measurable function g, which represents the evaluation of the decision y in the realisation ω. By making this choice, the evaluation of a possible solution y under a specific realisation ω is directly related the probability $P(\omega)$ that ω occurs. The overall quality of a solution is then taken to be the aggregation over all realisations ω.

On the other hand, there are contexts where particular realisations, corresponding to extreme occurrences, are of disproportional importance. In such settings, more complicated ways of evaluating a potential solution, such as the *conditional value at risk* (Dupačova and Polívka, 2007) can be preferable. In this article, we will focus on the choice (2.2) and defer consideration of other possible definitions for Φ to future work.

The scenarios approach mentioned in the introduction enters here as follows. Let $\mathcal{S}$ be a finite subset of a configuration space of parameters. These possible configurations $s \in \mathcal{S}$, as well as the probabilities p_s assigned to them, can either be inferred from past data that is used to derive probabilistic information, or generated by subjective analysis, as described in the introduction. We refer to elements of $\mathcal{S}$ as *scenarios*: each scenario corresponds to a possible configuration of parameters that might occur. In this way, we obtain a discretisation of (2.1) by setting $P = \sum_{s \in \mathcal{S}} p_s \delta_s$, so that (2.1) with the choice (2.2) becomes

$$\phi(P) = \phi\left(p_s \colon s \in \mathcal{S}\right) = \inf_{y \in \mathcal{Y}} \sum_{s \in \mathcal{S}} g(y; s) p_s \tag{2.3}$$

3. Methodology

Suppose that we had an oracle that predicts with certainty which scenario s will occur. Then we can always choose the best decision

$$y_s^* = \operatorname*{argmin}_y g(y; s) \tag{3.1}$$

under this scenario. The use of such point estimates as a base for decision-making is frequently taken. In this context, the decision making process takes the form of a two-step approach. In the first step, a point prediction is made, based on which, as a second step a prescriptive model is applied to establish the decisions to be made. The first step often relies on the use of a predictive model (which can be thought of as an 'oracle'), that

establishes which scenario is most likely to occur. Given this scenario s, an optimal decision y_s^* is obtained via a prescriptive model.

In reality, a perfect oracle does not exist, and this process will lead to incorrect predictions and therefore non-optimal decision being taken from time to time. In order to quantify this error, we introduce the *opportunity cost* of taking the decision associated to scenario s_1 when another scenario s_2 actually occurs. Denote this by $\delta(s_1 | s_2)$:

$$\delta(s_1 | s_2) = g\left(y_{s_1}^*; s_2\right) - g\left(y_{s_2}^*; s_2\right) \geq 0. \tag{3.2}$$

Since $\delta(s_1 | s_2) \neq \delta(s_2 | s_1)$ in general, we will symmetrise and define the *opportunity cost distance function* on $\mathcal{S}$ by

$$d(s_1, s_2) = \delta(s_1 | s_2) + \delta(s_2 | s_1), \qquad s_1, s_2 \in \mathcal{S}. \tag{3.3}$$

This introduces a notion of distance on the set $\mathcal{S}$ of scenarios. This distance function enables us to compare scenarios on a decisional basis. It is natural to now identify groups of scenarios which are close to each other with respect to this distance, since the decision associated to one scenario in this group will still be close to optimal for the others. Such groups also yield another way of estimating the risk associated to the predictions made by the oracle. Furthermore, in the special case of a linear problem with fixed recourse we have the following result:

PROPOSITION 3.1. *Suppose that the problem* (2.3) *is linear and with fixed resource. For each $s \in \mathcal{S}$ let $\mathcal{Y}_s^*$ denote the set of optimal solutions of the deterministic problem* (3.1) *corresponding to scenario s. If there exists a solution $\mathcal{Y}^* \in \bigcap_{s\in\mathcal{S}} \mathcal{Y}_s^*$ then y^* is also optimal for the stochastic problem.*

Proof. Suppose for a contradiction that there exists $\tilde{y} \in \mathcal{Y}_{\text{stoch}}$ (the feasible set for the stochastic problem) such

$$\sum_{s\in\mathcal{S}} g(\tilde{y}, \xi_s)\, p_s < \sum_{s\in\mathcal{S}} g(y^*, \xi_s)\, p_s. \tag{3.4}$$

Then there must exist at least one scenario $s \in \mathcal{S}$ with $g(\tilde{y}, \xi_s) < g(y^*, \xi_s)$. By optimality of y^* for the deterministic problem it follows that $\tilde{y} \notin \mathcal{Y}_s$. But this is a contradiction since then $\tilde{y} \notin \mathcal{Y}_{\text{stoch}} = \bigcap_{s\in\mathcal{S}} \mathcal{Y}_s$, the latter set equality being Theorem 4 in Birge and Louveaux (2011). □

COROLLARY 3.2. *Suppose that $C \subseteq \mathcal{S}$ is such that y^* is optimal for the deterministic problem with respect to all $s \in C$. Then the conditional expectation on C is also maximised by y^*.*

In particular, if we knew that a cluster actually has an optimal solution in common we could replace the entire cluster by one single representative without changing the solution. Thus, in a perfect clustering, Corollary 3.2 allows us to eliminate scenarios with the same associated decision. However, such a perfect agreement is unlikely to be the case in practice. While exact optimality will not often be observed in practice, Corollary 3.2 nevertheless serves as further motivation to find groups with mutually acceptable solutions.

In the remainder of this section, we will first detail how our general methodology is imbeded within a general decision support process under uncertainty (see subsection 3.1). In subsection 3.2 we give two example problems to which we will apply our methodology.

Mathematically, our technique of finding such clusters consists of defining a graph with vertex set $\mathcal{S}$, based on the notion of distance induced by d. The motivation for this approach lies in the existence of good methods for identifying clusters in a graph. In particular, we will apply Ncut (Shi and Malik, 2000) and its relaxation, spectral clustering; see von Luxburg (2007) for a survey. The technical aspects are explained in section 3.3 below.

3.1. **Decision-based clustering for decision support under uncertainty.** We now return to the general problem stated in (2.1). Commonly (Birge and Louveaux, 2011), one of two approaches is taken:

(A) The classical approach of predicting a particular outcome and then proscribing a solution accordingly, as described above. This corresponds to predicting a scenario $s \in \Omega$ and computing $\phi(\delta_s)$ as an approximation to $\phi(P)$. This approach has the advantage of being computationally cheap. As a downside, the error, i.e. the difference between $\phi(P)$ and $\phi(\delta_s)$ is not controlled.

(B) The stochastic programming approach: according to the probability measure P, sample scenarios $s_1, \ldots, s_N$ and attach probabilities $p_{s_1}, \ldots, p_{s_N}$. Then approximate $\phi(P)$ by $\phi(P_N)$ where

$$P_N = \sum_{k=1}^{N} p_k \delta_{s_k}. \tag{3.5}$$

This approach leads to a much better approximation: under certain assumptions about the sampling procedure, one can sometimes even get arbitrarily close to $\phi(P)$ by taking N large enough. On the other hand, the complexity is uncontrolled.

We propose the following approach, which will control both the complexity and the error incurred:

Step 1: Generate scenarios $s_1, ..., s_N$ as in (B).

Step 2: For each scenario s_j, calculate $\phi(\delta_{s_j})$ as in (A). In particular, we obtain a minimiser $y^*_{s_j}$ for each scenario s_j; that is

$$\Phi\left(y, \delta_{s_j}\right) \geq \Phi\left(y^*_{s_j}, \delta_{s_j}\right) \quad \forall\ y \in \mathcal{Y}. \tag{3.6}$$

Step 3: Compute opportunity costs of scenario s_i with respect to s_j, that is the loss incurred by optimising under the expectation that scenario s_i happens, when actually scenario s_j occurs:

$$\delta\left(s_i \,|s_j\right) = \Phi\left(y^*_{s_i}, \delta_{s_j}\right) - \Phi\left(y^*_{s_j}, \delta_{s_j}\right), \tag{3.7}$$

which induces a distance on the space of scenarios $\mathcal{S} = \{s_1, \ldots, s_N\}$ as in (3.3).

Step 4: Apply clustering methods in order to find groups of scenarios that are close to each other in terms of the optimal solution. Thus, we obtain a partition $C_1, ..., C_n$ of the space of scenarios $\mathcal{S}$.

As illustrated in Figure 1, the traditional approach to stochastic optimisation calls for the generation of a set of representative scenarios that are used to instantiate a scenario-based stochastic optimisation model, which is solved to obtain a solution. For example, as in the case of the sample-average approximation method (Kleywegt et al., 2002), this series of steps is repeated to evaluate probabilistic upper and lower bounds.

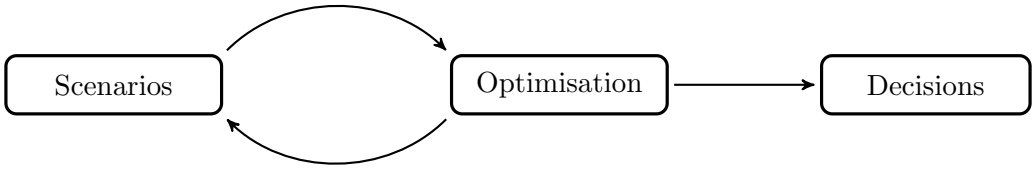

FIGURE 1. Traditional approach to stochastic optimisation.

One way to exploit this clustering is as follows. Choose one representative scenario $\sigma_j \in C_j$ for each cluster C_j and assign it a probability π_j. A natural example is given by

$$\pi_j = \sum_{s \in C_j} p_s. \tag{3.8}$$

Then, find $\Phi\left(\sum_{k=1}^{n} \pi_k \delta_{\sigma_k}\right)$ and use this as an approximation for $\phi(P)$. Thus, we obtain an approximation that is computationally much cheaper than (B) (since in general the number of clusters n is much smaller than the number of scenarios N), but at the same time we are able to control the error incurred in terms of the diameters of the clusters. Alternatively, one can refine approach (A) by choosing a point estimate for each cluster, rather than for each scenario.

Our approach works well for choices of Φ where the error incurred by approach (A) is too large, whereas approach (B) is too computationally expensive. By following the four steps outlined above, we obtain a control on both the error and the computational complexity, see Figure 2. In fact, we can trade off complexity (the number of clusters) against accuracy (the diameter of clusters), in order to obtain an optimal approximation.

In particular, it should be noted that because we are only solving the deterministic version of (2.1), we can consider a much larger number of scenarios than the stochastic approach would be able to handle.

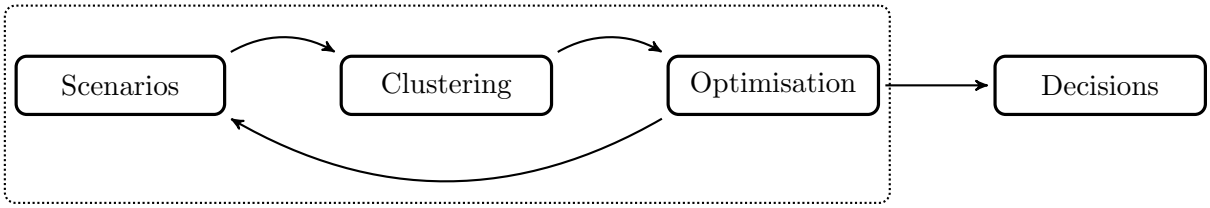

FIGURE 2. Embedding decision-based clustering into the decision support process

3.2. **Examples.** We have chosen two stochastic optimisation problems on which to test our approach, one modelling the design of a transport network and one from fleet planning.
Transport network design. When designing a transport network, two types of decisions must be taken Crainic et al. (2014): first, one chooses the structure of the network (*design decisions*) and secondly how to use this network to perform the operational activities considered (*flow decisions*).

Our specific problem was first considered in Crainic et al. (2016). Here, the design decisions must be taken before the stochastic parameters are known. More precisely, consider a directed graph $G = (N, A)$ and a set of commodities K. For each scenario $s \in \mathcal{S}$, the stochastic parameters are given by the demands d_i^{ks} of the quantity of commodity k to be transported to vertex $i \in N$ and the capacity $u_{i,j}^s$ of edge (i, j).

For each edge $e = (i, j) \in A$, the design decision corresponds to choosing whether to open $e = (i, j)$ at a fixed cost y_{ij} or not. This decision must be taken before the scenario that actually occurs is known. Once the design decision has been taken, the scenario is revealed and the flow decisions must be taken. That is, we must choose how many units of commodity k to transport across edge (i, j), at a unit cost of c_{ij}^k. The goal of the program is to satisfy all of the demands while minimising the expected total cost incurred. Denoting by $y_{i,j} \in \{0, 1\}$ the choice of opening edge (i, j) and $x_{i,j}^{k,s}$ the quantity of commodity k to be transported across (i, j) in the case of scenario s, we obtain the following mathematical formulation:

$$\min \sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{s\in\mathcal{S}} p_s \left(\sum_{k\in K} \sum_{(i,j)\in A} c_{ij}^k x_{ij} \right) \tag{3.9}$$

$$\text{s.t.} \sum_{j\in N^+(i)} x_{ij}^{ks} - \sum_{j\in N^-(i)} x_{ji}^{ks} = d_i^{ks} \qquad \forall\, (i, k, s) \in N \times K \times \mathcal{S} \tag{3.10}$$

$$\sum_{k\in K} x_{ij}^{ks} \le u_{ij}^s y_{ij} \qquad \forall \big((i, j), s\big) \in A \times \mathcal{S} \tag{3.11}$$

$$y_{ij} \in \{0, 1\}, \quad x_{ij}^{ks} \in [0, \infty) \qquad \forall \big((i, j), k, s\big) \in A \times K \times \mathcal{S}. \tag{3.12}$$

Biweekly fleet planning. Our second test problem concerns the fleet-sizing problem faced by a freight carrier over a two-week horizon where the loads for the first week are known (Topaloglu, 2018). The decisions are for the first week are the number of vehicles available at each terminal and the number of vehicles moving between each origin and destination. The decisions for the second week are similar, but the vehicle supply at each terminal is determined by the decisions of the first week. If the vehicle supply of a location at the end of the second week is different than what it was at the beginning of the first week, then a penalty is incurred. We introduce the following notation: x_{ij} and $\tilde{x}_{ij}$ denote number of empty and loaded of vehicles respectively, moving terminal $i$to j during the first week. The variables z_i represent the number of vehicles deployed at terminal i. The number of vehicles at terminal i at the beginning of the second week is given by $\tilde{z}_i$. If the vehicle supply at location j at the end of the second week is below z_j , then this imbalance is penalized by a cost of p_j per unit shortage. ξ_{ij} denotes the (random) number of loads that need to be carried from terminal i to j in the second stage. The mathematical formulation of the problem is then given by

$$\min \sum_{i\in\mathcal{L}} v_i z_i + \sum_{i,j\in\mathcal{L}} (r_{ij} x_{ij} + \tilde{r}_{ij} \tilde{x}_{ij}) + Q\,(z, \tilde{z}) \tag{3.13}$$

$$\text{s.t.} \sum_{j\in\mathcal{L}} (x_{ij} + \tilde{x}_{ij}) - z_i = 0 \qquad \forall\, i \in \mathcal{L} \tag{3.14}$$

$$\sum_{i\in\mathcal{L}}(x_{ij}+\tilde{x}_{ij})-\tilde{z}_i=0 \qquad \forall\, i\in\mathcal{L} \tag{3.15}$$

$$x_{ij}\leq u_{ij} \qquad \forall\,(i,j)\in\mathcal{L}^2 \tag{3.16}$$

where the function Q is defined by

$$Q\left(z,\tilde{z},\xi\right)=\min\sum_{i,j\in\mathcal{L}}(r_{ij}y_{ij}+\tilde{r}_{ij}\tilde{y}_{ij})+\sum_{j\in\mathcal{L}}p_jw_j \tag{3.17}$$

$$\text{s.t.}\sum_{j\in\mathcal{L}}(y_{ij}+\tilde{y}_{ij})=\tilde{z}_i \qquad \forall\, i\in\mathcal{L} \tag{3.18}$$

$$\sum_{i\in\mathcal{L}}(y_{ij}+\tilde{y}_{ij})+w_j-\tilde{w}_j=z_j \qquad \forall\, j\in\mathcal{L} \tag{3.19}$$

$$y_{ij}\leq\xi_{ij} \qquad \forall\,(i,j)\in\mathcal{L}^2. \tag{3.20}$$

3.3. **Graph clustering.** We conclude this section by giving some details on the graph clustering techniques we are proposing to use. There are two key questions: 1) how to define the graph on the scenario space and 2) based on this graph, how to partition the scenarios into clusters.

Given a distance function d on $\mathcal{S}$, there are two standard ways to construct *affinity graphs* with vertex set $\mathcal{S}$: namely we define $G=(\mathcal{S},E)$ where the edge set E is one of the following:

(1) $E=\{d(s,t)\colon a_{st}<\epsilon\}$ for some (small) parameter $\epsilon>0$. That is, two scenarios s,t are connected if the opportunity cost between s and t is smaller than ϵ. This graph is known as the *ϵ-neighbourhood graph*.

(2) Let $s\rightsquigarrow t$ if $d(s,t)$ is one of the M smallest elements of $\{d(s,u)\colon u\neq s\}$ and $s\sim t$ if $s\rightsquigarrow t$ and $t\rightsquigarrow s$. The edge set E is then defined to be $E=\{(s,t)\in\mathcal{S}\times\mathcal{S}\colon s\sim t\}$. We will call this the *(symmetrised) M-nearest neighbour graph*.

In both cases, the graph is a way of encoding *affinity* between scenarios: two scenarios s,t are connected if the opportunity costs between s and t are small. Moreover, there is always a parameter (ϵ or M) that can be adjusted in order to regulate how much affinity is rewarded.

Once we have constructed our affinity graph (either ϵ-neighbourhood graph or M-nearest neighbour graph), we can analyse it in order to identify structure in the scenarios. Our focus in this paper will be on clustering algorith, but we note in passing that other analyses are also possible, including finding maximal graph cliques (Tsukiyama et al., 1977) and analysing connected components.

The goal of our clustering is to partition the scenario space $\mathcal{S}$ into clusters $C_1,...,C_n$ such that the diameters of the C_j are as small as possible. In order to simplify the presentation, we will assume that the choice of affinity graph and parameter (ϵ or M) has been made and let G denote the affinity graph chosen for analysis.

The simplest approach to this would be to solve the *min-cut problem* described in Stoer and Wagner (1997). However, this method tends to simply separate out single elements from the vertex set. Thus, one needs to incorporate into the objective function the requirement that all clusters be reasonably large. Examples of this approach are the RatioCut (Hagen and Kahng, 1992) and Ncut (Shi and Malik, 2000) methods. For a problem with a relatively

small number of scenarios, Ncut yields very good results. However, these extensions are NP hard to solve (Wagner and Wagner, 1993) and hence the computational complexity grows quickly with the number of vertices, or scenarios.

All of the min-cut problems described above can be expressed in terms of discrete minimisation problems involving graph Laplacians. Therefore, it is natural to consider their continuous relaxations. These relaxations can be written in terms of various graph Laplacians. In addition to NCut itself, we utilise a relaxation of the NCut problem, the so-called *normalized spectral clustering* (Shi and Malik, 2000; Ng et al., 2002):

Algorithm: Normalised Spectral Clustering.

(1) Compute the random walk Laplacian L_{rw}.
(2) Compute the eigenvalues $u_1, \ldots, u_k$ corresponding to the k lowest eigenvalues of L_{rw}
(3) For each $s \in \mathcal{S}$ define $y_s \in \mathbb{R}^k$ by $y_s(j) = u_j(s)$
(4) Perform the k-means algorithm to partition the y_s into k clusters $\Gamma_1, \ldots, \Gamma_k$
(5) return the partition $C_1, \ldots, C_m$ on $\mathcal{S}$ defined by $s \in C_j$ if and only if $y_s \in \Gamma_j$.

4. Numerical results

In order to test our approach, we have applied it to the two examples described in section 3.2. In this section, we summarise the preliminary results we have obtained. We have found that our method works well for complex problems with limited computational resources or time.

4.1. **Stochastic network design.** We have applied our methodology to 20 instances of the stochastic network design problem, each having 128 scenarios. Using the Ncut algorithm described above we have clustered the scenario space into groups of between 2 and 15 clusters. We then applied the group subproblem methodology explained in Sandikci et al. (2013). Thies yields a lower bound, and the solution of the group subproblem defines a feasible solution to the original problem, which leads to an upper bound. At the same time, we solved the full stochastic problem using CPLEX with a time limit of 10 hours, which led to another set of upper and lower bounds. We evaluated the effectiveness of our approach by calculating the relative difference between the lower bound from the group subproblem and the lower bound from the full problem.

These relative differences are listed in Table 1 below, aggregated across the instances by the number of clusters. The *cluster bound gap* denotes the relative difference between the lower bound from the full model and the lower bound from the group subproblem, whereas the *cluster primal gap* is calculated as the relative difference between the upper bound from the group subproblem and the upper bound from the full model. Thus, in both cases a negative value implies an improvement in performance of our clustered approach. Finally, we have also computed the difference in time taken between the clustered model and the full model. As can be seen in Table 1, our approach is faster.

	cluster bound gap		**cluster primal gap**		**time difference**	
	mean	var	mean	var	mean	var
clusters						
2	-0.009716	0.000929	-0.044691	0.017231	-4625.80	1.268810e+08
3	-0.018039	0.002010	-0.067526	0.025848	-4374.25	9.673558e+07
4	-0.021404	0.002775	-0.092313	0.039781	-3306.70	8.473449e+07
5	-0.023190	0.003024	-0.089347	0.047737	-2292.30	1.139200e+08
6	-0.024780	0.003174	-0.088608	0.047761	-1898.25	1.704076e+08
7	-0.026365	0.003577	-0.089912	0.049885	-965.00	2.228125e+08
8	-0.024229	0.003218	-0.089774	0.049921	-2150.20	2.615600e+08
9	-0.017766	0.001179	-0.089775	0.049932	-1666.85	3.202642e+08
10	-0.018919	0.001341	-0.089793	0.049922	-1137.65	3.746308e+08
11	-0.012263	0.000485	-0.057347	0.019306	-1820.75	3.769358e+08
12	-0.012191	0.000462	-0.048259	0.009202	-2795.10	3.768468e+08
13	-0.009643	0.000369	-0.039941	0.008548	-4169.70	3.236817e+08
14	-0.009015	0.000358	-0.040060	0.008539	-5158.70	2.985408e+08
15	-0.007975	0.000344	-0.039514	0.008580	-6497.45	2.432807e+08

TABLE 1. Summary performance statistics for our approach in the network design case

4.2. **Biweekly fleet planning.** For biweekly fleet problem, we have tested four instances of the small problem and compared the group subproblem bound (see above) with the deterministic approximation (DA) described in Topaloglu (2018).

	cluster bound gap	
	mean	var
clusters		
2	-0.063961	0.000162
3	-0.071368	0.000058
4	-0.082579	0.000099
5	-0.086557	0.000091
6	-0.092331	0.000078
7	-0.097669	0.000084
8	-0.101370	0.000099
9	-0.106733	0.000092
10	-0.109695	0.000068
11	-0.113290	0.000068
12	-0.117053	0.000055
13	-0.120289	0.000054
14	-0.123160	0.000052

TABLE 2. Performance statistics comparing the group subproblem bound applied to our clustering method with the deterministic approximation.

5. CONCLUSION

We have introduced a new methodology that enables a large number of scenarios to be efficiently analysed. Specifically, this methodology allows to identify scenarios which are close on a decisional basis. Such groupings of scenarios can then be applied to gain a better understanding of how stochastic phenomena can impact the decision-making process.

We have shown that these clusters can be used to efficiently derive lower and upper bounds in a context of performing stochastic optimisation. When applied to the case of stochastic network design, we observe that, under computational time constraints, these bounds are better and can be obtained more quickly when compared to solving the considered problems directly using CPLEX, an efficient and widely used commercial optimisation software.

Going forward, various avenues of research appear to us interesting to investigate. First, from a computational point of view, calculating the opportunity cost matrix may be costly if the number of scenarios is large, or, if the decisional problem studied is complex to solve. The investigation of how parallel computing can be used to speed up the building of this matrix is thus an interesting avenue of research to explore. Second, finding how scenarios, that are used in specific decision-making contexts, are related to one another is also a general open question that should be studied. Towards this end, we will also investigate alternative decision- and scenario-based distance functions and consider directed graph clustering methods that can be applied to asymmetric distance functions for the proposed methodology. Finally, we have used the expected value in order to evaluate the quality of the decisions to be made when facing uncertainty. In contexts where extreme cases are of disproportional importance, other functionals such as the conditional value at risk may

be more suitable to apply. Exploring how the developed clustering method can be used in such cases will also be studied.

References

Bell, D. E. (1982). Regret in decision making under uncertainty. *Operations Research*, 30(5):961–981.

Birge, J. R. and Louveaux, F. V. (2011). *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media, second edition.

Bloom, N., Bond, S., and van Reenen, J. (2007). Uncertainty and investment dynamics. *Review of Economic Studies*, 74:391–415.

Borgonovo, E., Cappelli, V., Maccheroni, F., and Marinacci, M. (2018). Risk analysis and decision theory: A bridge. *European Journal of Operational Research*, 264:280–293.

Crainic, T., Hewitt, M., Maggioni, F., and Rei, W. (2016). Partial Benders decomposition strategies for two-stage stochastic integer programs. *CIRRELT*, 2016-37.

Crainic, T., Hewitt, M., and Rei, W. (2014). Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research*, 43:90 – 99.

Crainic, T. G., Fu, X., Gendreau, M., Rei, W., and Wallace, S. W. (2011). Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124.

Devaraj, S. and Kohli, R. (2003). Performance impacts of information systems: Is actual usage the missing link? *Management Science*, 49(3):273–289.

Dupačova, J. and Polívka, J. (2007). Stress testing for VaR and CVaR. *Quantitative Finance*, 7(4):411–421.

Edmunds, A. and Morris, A. (2000). The problem of information overload in business organisations: a review of the literature. *International Journal of Information Management*, 20:17–28.

Godet, M. (2000). The art of scenarios and strategic planning: Tools ans pitfalls. *Technological Forecasting and Social Change*, 65:3–22.

Hagen, L. and Kahng, A. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, 11(9):1074 – 1085.

Higle, J. L. and Sen, S. (1991). Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669.

Høyland, K., Kaut, M., and Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3):169–185.

King, A. J. and Wallace, S. W. (2012). *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer Science & Business Media.

Kleywegt, A. J., Shapiro, A., and de Mello, T. H. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502.

Lium, A.-G., Crainic, T. G., and Wallace, S. W. (2009). A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144 – 157.

Löhndorf, N. (2016). An empirical analysis of scenario generation methods for stochastic optimization. *European Journal of Operational Research*, 255(1):121–132.

Mackelpranga, A. W. and Malhotra, M. K. (2015). The impact of bullwhip on supply chains: Performance pathways, control mechanisms, and managerial levers. *Journal of Operations Management*, 36:15–32.

Melville, N., Kraemer, K., and Gurbaxani, V. (2004). Review: Information technology and organizational performance: An integrative model of it business value. *MIS Quaterly*, 28(2):283–322.

Metters, R. (1997). Quantifying the bullwhip effect in supply chains. *Journal of Operations Management*, 15:89–100.

Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14, pages 849 – 856. MIT Press.

Ouyang, Y. and Li, X. (2010). The bullwhip effect in supply chain networks. *European Journal of Operational Research*, 201(3):799–810.

Peter, M. K. and Jarratt, D. G. (2015). The practice of foresight in long-term planning. *Technological Forecasting and Social Change*, 101:49–61.

Pownuk, A. and Kreinovich, V. (2018). *Combining Interval, Probabilistic, and Other Types of Uncertainty in Engineering Applications*, volume 773 of *Studies in Computational Intelligence*, chapter 7. Decision Making Under Uncertainty, pages 157–190. Springer, Cham.

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2018). Accelerating the benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization*, 28(1):875–903.

Sandikci, B., Kong, N., and Schaefer, A. J. (2013). A hierarchy of bounds for stochastic mixed-integer programs. *Math. Program., Ser. A*, 138:253 272.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888 – 905.

Stoer, M. and Wagner, F. (1997). A simple min-cut algorithm. *J. ACM*, 44(4):585 – 591.

Topaloglu, H. (2018). Stochastic programming data sets.

Trieu, V.-H. (2017). Getting value from business intelligence systems: A review and research agenda. *Decision Support Systems*, 93:111–124.

Tsukiyama, S., Ide, M., Ariyoshi, I., and Shirakawa, I. (1977). A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505517.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395 – 416.

Wagner, D. and Wagner, F. (1993). Between min cut and graph bisection. In *Proceedings of the 18th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 744 – 750. Springer.