NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Dynamic exploration designs for graphical models using clustering with applications to petroleum exploration

by

Gabriele Martinelli and Jo Eidsvik

PREPRINT
STATISTICS NO. 9/2012

This report has URL http://www.math.ntnu.no/preprint/statistics/2012/S9-2012.pdf
Jo Eidsvik has homepage: http://www.math.ntnu.no/∼joeid
E-mail: joeid@stat.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Dynamic exploration designs for graphical models using clustering with applications to petroleum exploration

Gabriele Martinelli and Jo Eidsvik

Department of Mathematical Sciences, NTNU, Norway

Abstract

The paper considers the problem of optimal sequential design for graphical models. Oil and gas exploration is the main application. Here, the outcomes at prospects or reservoir units are highly dependent on each other. The joint probability model for all node variables is considered known. As data is collected, this probability model is updated. The sequential design problem entails a dynamic selection of nodes for data collection, where the goal is to maximise utility, here defined via entropy or total expected profit. With a large number of nodes, the optimal solution to this selection problem is not tractable. An approximation based on a subdivision of the graph is considered. Within the small clusters the design problem can be solved exactly. The results on clusters are combined in a dynamic manner, to create sequential designs for the entire graph. The merging of clusters also gives upper bounds for the actual utility. Several synthetic models are studied, along with two real cases from the oil and gas industry. In these examples Bayesian networks or Markov random fields are used. The sequential model updating and data collection strategies provide useful guidelines to policy makers.

1 Introduction

Our interest is a sequential selection problem over dependent variables. The main motivation is to construct policies for oil and gas exploration, where the outcomes at prospects are dependent by spatial proximity or by common geological mechanisms. The probability of success for any exploration well is then highly influenced by the outcomes at other prospects.

More generally the challenge is to construct an optimal dynamic design of nodes in a graph. For instance, in the situation with a Bayesian Network (BN) or a Markov Random Field (MRF) we evaluate which variables are most useful to observe. We assume a fixed probability model a priori. As we acquire data at nodes in the BN or the MRF, the original probability distribution is updated, according to Bayes rule. Relevant design questions are then: Which nodes are more informative? Which sequence of nodes gives the best policy? In the petroleum industry drilling wells is extremely costly, and getting the right information is critical.

At each stage of the dynamic strategy, we choose to observe one additional variable, or quit the search. If we acquire data at a node, we incorporate the observation in the current (a priori) model to compute the updated (a posteriori) model. For the next stage, the updated model serves as a prior model, and so on. The sequential decisions account for two aspects: i) the immediate profit in terms of monetary units or information gain by knowing the current variable, and ii) the expected future benefits induced by the predictive capacity, conditional on the current variable. These two aspects are combined in a utility function. If the expected utility of choosing one more node is too small, we stop collecting data. The trade off between i) and ii) is related to more general explore

or exploit problems in decision making. An oil and gas company may want to target first the most lucrative prospects, but it is also important to know the key variables, which give us the chance to make better, informed, decisions at the later stages. The future values in ii) then play an important role in the utility function.

With our focus on oil and gas exploration we note some similarities and differences with common spatial design problems, e.g. Shewry and Wynn (1987), Le and Zidek (2006) and Zidek and Zimmerman (2009). The most common problem treated in the literature is to allocate a fixed number of monitoring stations to improve overall predictive performance in some sense. The selection is thus done in the static manner, not allowing the decision maker to modify her choices after observing the outcomes at the previously selected spatial sites. In this paper we consider the dynamic decision problem, with one observation at a time and the ability to make sequential decisions. Moreover, in spatial design problems the model is typically Gaussian. Our paper is new in the sense that it studies design for graphical models with discrete outcomes at all nodes.

Our sequential design problem is a discrete optimization problem which is in theory solved via dynamic programming (DP). This method defines a forward-backward algorithm that constructs the optimal sequences and the expected utility. Bickel and Smith (2006) present a DP algorithm tailored for our sequential design problem with dependent oil and gas prospects. However, their approach is not applicable when the number of variables gets too large. For more than, say, ten variables, we must instead look for approximate strategies. The appropriate solution seems to be very case-specific. See e.g. Powell (2007) for more background. Various heuristic approaches are important for special applications, but it is very difficult to assess the properties of these solutions. For graphical models it seems natural to utilise the structure. One approach is to split the original graph in several disjoint clusters. This idea was originally presented in Brown and Smith (2012). They next solved the DP exactly for the clusters, and combined the results to get approximations for the expected utilities on the full-size graph. The approach also allows an upper bound on the utility, indicating the quality of the approximation.

Our main contribution in this paper is to use the clustering strategies for graphs to construct sequential designs for BNs and MRFs. A critical element in the method is to compute the cluster-wise Gittins index. This extends the original index pioneered in statistics by Gittins (1979) and Whittle (1980) for so-called bandit problems, studied by Benkerhouf et al. (1992) and Glazebrook and Boys (1995) for oil and gas exploration problems. We consider the sensitivity of cluster orientation and size, and various levels of approximation in the Bayes updating scheme. We use utility functions based on entropy and more traditional cost/revenue aspects. For the situation with dependent oil and gas prospects, the resulting designs can work as a road map for the exploration company. In this way we combine statistical models and Bayesian updating with decision making to create policies. Our focus is on oil and gas resources applications, but similar methods are relevant for e.g. machine scheduling (Abdul-Razaq and Potts, 1988), medical treatments selection (Claxton and Thompson, 2001), subset selection problems and more generic search problems.

The paper develops in the following way: in Section 2 we give the main ideas about sequential design, in Section 3 we discuss how the splitting in clusters can help in building approximate strategies, in Section 4 we provide results on synthetic examples, in Section 5 we show results on real case studies.

2 Sequential design

A sequential strategy is illustrated in Figure 1 for the context of petroleum exploration. Here, we initially choose to drill one of three prospects, or nothing. If we start by drilling prospect 3, the

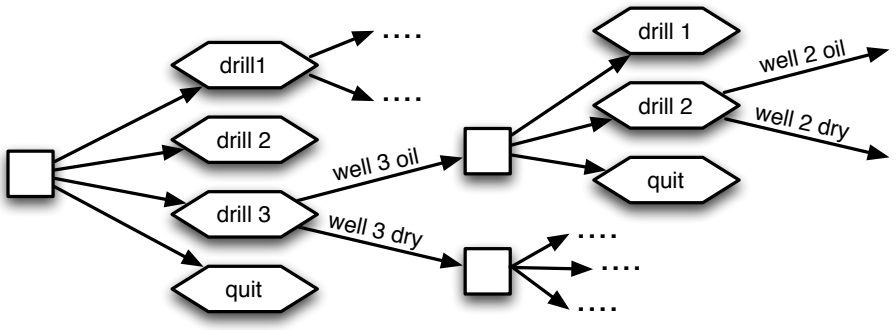

Figure 1: Decision tree for a simple 3-nodes discrete example with two possible outcomes (*oil* or *dry*) per node.

design criterion for the next stage depends on the outcome of prospect 3. The decision is then to choose among prospect 1 and 2, or quit.

Similarly, the sequential design problem we consider here entails a selection of nodes, one at a time, to maximize a utility function. We first introduce the statistical notation and assumptions required to frame this sequential design problem. We next outline the theoretical solution given by DP. A small example is then used to illustrate the sequential strategies resulting from different utility functions.

2.1 Notation and modeling assumptions

Consider N nodes, and let $x_i \in \{1, \ldots, k_i\}$, $i = 1, \ldots, N$ denote the discrete random variables. Without loss of generality, we assume $k_i = k$ possible states for all nodes i. In Figure 1, $k = 2$ with oil or dry outcomes. We represent the probabilistic structure for $\mathbf{x} = (x_1, \ldots, x_N)$ via a graph. For a BN defined by a directed acyclic graph the joint distribution is

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i | x_{\mathrm{pa}(i)}), \tag{1}$$

where $\mathrm{pa}(i)$ denotes the parent set of node i, which is empty for the top nodes. Undirected graphs are defined via the full conditionals over a neighborhood, or, by the Hammersley-Clifford theorem, via a joint distribution over clique potentials. For a first-order MRF (Besag, 1974) we use:

$$p(\mathbf{x}) \propto \exp\left\{ \beta \cdot \sum_{i \sim j} \mathbb{I}(x_i = x_j) + \sum_{i=1}^{N} \alpha_i(x_i) \right\}, \tag{2}$$

where $i \sim j$ denotes neighboring lattice nodes (north, east, south, and west). The parameter β imposes spatial interaction, while the $\alpha_i(x_i)$ terms include prior preferences about states at node i.

We assume known, fixed, statistical model parameters in $p(\mathbf{x})$, such as β and $\alpha_i(x_i)$ in (2) and the conditional probabilities in (1). Associated with the probabilistic model we can of course compute several attributes that are important for design purposes. Assuming that we know the revenues or cost, denoted r_i^j, for outcomes $x_i = j$, the decision value (DV) is $DV(i) = \max(0, \sum_{j=1}^{k} r_i^j p(x_i = j))$, $i = 1, \ldots, N$. This DV is useful for decision making. It is non-zero only when the expected profit is positive. The entropy (disorder) is defined by $H = -\sum \log(p(\mathbf{x}))p(\mathbf{x}) = -E(\log p(\mathbf{x}))$, and the reduction in entropy is often used for design purposes, see Wang and Suen (1984).

In our sequential design situation, we rely on the ability to extract the marginal probabilities at all nodes, and to update the probability distributions when evidence is collected. Since we are going to update the model at each stage of the sequential strategy, for many different kinds of evidence, we require these computations to be reasonably fast. For BNs the updating of probabilities can be done effectively by the junction tree algorithm (Lauritzen and Spiegelhalter (1988)). MRFs can similarly be updated by forward-backward algorithms, see e.g. Reeves and Pettitt (2004) and Tjelmeland and Austad (2012).

Assume we can acquire data at one node in the graph, and incorporate the outcome to get a posterior distribution. For the next stage, this updated distribution serves as a prior distribution. We can then select another node, acquire information, update the probabilities, and so on. The sequential design of nodes is constructed by optimizing the expected utility, which means that we integrate over all possible data when finding the optimal sequence. In our case, the utility is based on monetary profits or entropy reduction. One could of course imagine other selection criteria here. Minimum entropy entails a dynamic design that attempts to stabilize or minimize the uncertainty in the graph.

Let ω_i be the observable or evidence in node $i = 1, \ldots, N$. If node i is not yet observed, we set $\omega_i = -$. If we choose to observe node i, ω_i is the actual outcome of the random variable x_i at this node. For instance, in a petroleum example, $\omega_i = 1$ can mean that prospect i has been drilled and found dry, $\omega_i = 2$ if found gas, and $\omega_i = 3$ if oil. A priori, before acquiring any observables, we have $\boldsymbol{\omega} = \boldsymbol{\omega}_0 = (-, \ldots, -)$. When we observe nodes, we put the outcomes at the corresponding indices of the vector $\boldsymbol{\omega}$. Say, if node 2 is selected first, and observed in state $\omega_2 = x_2 = 1$, we set $\boldsymbol{\omega} = (-, 1, -, \ldots, -)$. At each stage, one more entry of $\boldsymbol{\omega}$ is assigned. The posterior that is updated at every stage of the sequential design is generically denoted by $p(\mathbf{x}|\boldsymbol{\omega})$, with marginals $p(x_i = j|\boldsymbol{\omega})$, $i = 1, \ldots, N$, $j = 1, \ldots, k$. Since we get perfect information about the selected node variables, we get $p(x_i = j|\boldsymbol{\omega}) = 0$ or 1 if node i is already observed.

In our setting it is important to monitor the design criterion or utility at all stages of sequential conditioning. When we get evidence $\boldsymbol{\omega}$, the entropy is reduced, so that $H(\boldsymbol{\omega}) - H \leq 0$. For the DV we have $\sum_{\boldsymbol{\omega}} DV(i|\boldsymbol{\omega})p(\boldsymbol{\omega}) \geq DV(i)$, where the probabilities for the DVs in the sum are conditional on the evidence $\boldsymbol{\omega}$. This entails that the pre-posterior DV is always larger than the prior value, and the value of information is always non-negative, see Bhattacharjya et al. (2010). The sequential design will be guided by immediate entropy reduction or gain in monetary value, as well as the expected future impact an observable can have.

2.2 Dynamic Programming for Sequential Design

The sequential design procedure forms a decision tree, where a fork represents a decision to choose a node (or quit), and each branch points to the future decisions, and the conditional utilities, depending on the outcome of the chosen node (Figure 1). See also Cowell et al. (2007), Chapter 8. We next present the method of DP to solve the sequential design problem. This algorithm computes the utilities of all possible sequential designs, and then picks the most lucrative sequence.

We first consider expected profit as utility function. This criterion is relevant for the petroleum examples with N prospects to explore and hopefully produce. Let $v(\boldsymbol{\omega})$ represent the expected revenues, i.e. future cash flows, given that we are in observation state $\boldsymbol{\omega}$. Initially, the vector of observables is empty: $\boldsymbol{\omega}_0 = \{-, -, \ldots, -\}$, and the value is $v(\boldsymbol{\omega}_0)$. DP computes $v(\boldsymbol{\omega}_0)$ and finds the associated optimal sequential design.

At the first stage we select the optimal node i among all nodes $\mathcal{N}$, or quit. The expected initial

value becomes

$$v(\boldsymbol{\omega}_0) = \max_{i \in \mathcal{N}} \left\{ \sum_{j=1}^{k} p(x_i = j) \left[r_i^j + \delta \max_{s \in \mathcal{N}/i} \left\{ \sum_{l=1}^{k} p(x_s = l | x_i = j)(r_s^l + \ldots), 0 \right\} \right], 0 \right\},$$

where the second and the subsequent maximizations are over nodes not yet considered in the sequential strategy. Here, δ is a discounting factor. In practice, a δ near 1 encourages learning the dependent model, while a smaller δ means that we choose the bigger DVs at the early stages. Note that the expected value contains immediate profit (r_i^j) and a continuation value (CV) with conditioning on the outcome of variable $x_i = j$ in the selected node. For short, we can write the expected revenues by starting at node i by

$$v_i(\boldsymbol{\omega}) = \sum_{j=1}^{k} p(x_i = j | \boldsymbol{\omega}) \left[r_i^j + \delta \cdot v(\boldsymbol{\omega}_i^j) \right], \tag{3}$$

where $\boldsymbol{\omega}_i^j = \{\boldsymbol{\omega}_{\text{-}\mathbf{i}}, \omega_i = j\}$ and $v(\boldsymbol{\omega}_i^j)$ is the CV of the state $\boldsymbol{\omega}_i^j$, i.e. $v(\boldsymbol{\omega}_i^j) = \max_{l \neq i}\{0, v_l(\boldsymbol{\omega}_i^j)\}$. If we know the outcomes at all nodes, the CV is $v(\cdot, \cdot, \ldots, \cdot) = 0$. This forms the starting point of DP, which proceed backwards, one step at a time, extracting the solutions for all sequences. We show an example in the next section.

As suggested in e.g. Weber et al. (2000) and Krause and Guestrin (2009), the reduction in entropy is useful for design. Let $H(\boldsymbol{\omega})$ be the (conditional) entropy with current evidence $\boldsymbol{\omega}$. We construct a sequential design based on

$$\Delta H(\boldsymbol{\omega}_i^j) = H(\boldsymbol{\omega}) - H(\boldsymbol{\omega}_i^j),$$

i.e. the reduction in entropy caused by additionally observing $x_i = j$ at one stage in the strategy. This entropy reduction can be computed efficiently utilizing fast updates of BNs and MRFs, and the conditional properties of entropy. We introduce a 'price' P_i of observing node i. This is assumed constant for any outcome of node i. Similar to what we did for the profit-based utility, we set the CV as the possible future reductions in entropy brought by the new observation. The expected utilities when including node i in the design becomes

$$v_i(\boldsymbol{\omega}) = \sum_{j=1}^{k} p(x_i = j | \boldsymbol{\omega}) \left[\Delta H(\boldsymbol{\omega}_i^j) - P_i + \delta \cdot v(\boldsymbol{\omega}_i^j) \right], \quad i = 1, \ldots, N. \tag{4}$$

The decision maker selects the node with the highest $v_i(\boldsymbol{\omega})$, i.e. the most informative part of the graph. If no nodes contribute with positive values, we quit the search. This means that the price P_i exceeds the expected immediate gain and future information reduction. When there are no more nodes to observe, the CV is 0. Similar to the situation above, the DP constructs the optimal sequential designs of nodes, and computes the associated reductions in entropy. An example is presented in the next section.

Note that the current situation with sequential decisions can also be phrased as a Markov Decision Process, where the generic state of the system develops as a function of the actions at each stage, see e.g. Puterman (2005). No matter how we interpret the sequential design problem, the computational cost grows exponentially with the number of nodes N. In the example below we construct optimal selection strategies among eight nodes. For graphs much larger than this, exact DP is not possible, and we outline the new approximate strategies in Section 3.

2.3 A motivating example

We now present the DP strategies driven by the cost/revenue utility function and entropy on a small example. The BN case study is shown in Figure 2. Here, the eight leaf nodes can be observed, $\{1A, 2A, \ldots, 5C\}$, while the remaining six auxiliary nodes, $\{K, P1, \ldots, P5\}$, impose the desired (causal) dependency structure in the BN (See Section 5). The goal is to determine where to observe first, and which would be the consequent choice, given data at the first node, and so on. We assume the initial probability structure of the BN is fixed. Each node has binary outcomes ($k = 2$). Inspired by the petroleum exploration, we refer to these two by 'oil' and 'dry'.

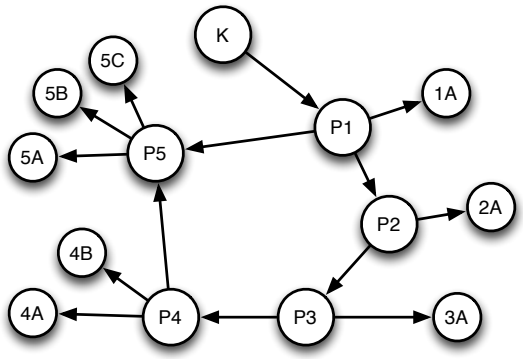

Figure 2: Example used in Section 2.3. We can collect data in the leaf nodes. By Dynamic Programming (DP) we construct optimal sequential designs that maximise expected utility.

The main input parameters for this example are given in Table 1. Based on cost/revenues only two nodes, 3A and 4A, have positive intrinsic value (IV) or marginal expectation $E(x_i)$. The DV is 0 when this value is non-positive. Thus, a naive decision maker, looking for profit, and ignoring the dependence between nodes, would forget about six of the prospects. The naive value of the field is $661 + \delta \cdot 514 = 1170$ for a specified discounting of $\delta = 0.99$. An optimal decision maker, using (3) and (4), would account for the ultimate consequences of the actions. Since there is much dependence in the BN model, the output will be quite different from the naive approach. Results

Prospect	1A	2A	3A	4A	4B	5A	5B	5C
$p(x_i = 0)$	0.44	0.46	0.48	0.61	0.70	0.40	0.48	0.48
$p(x_i = 1)$	0.56	0.54	0.52	0.39	0.30	0.60	0.52	0.52
Entropy reduction	0.6859	0.6899	0.6920	0.6682	0.6129	0.6743	0.6922	0.6922
Costs	3000	900	2400	1800	600	1500	3600	2100
Revenues	1368	707	3443	4151	1321	943	3254	1887
$E(x_i)$	-554	-32	661	514	-19	-41	-20	-18

Table 1: Input parameters for the example in section 2.3: Marginal probabilites, marginal entropy reductions and monetary parameters.

are shown in Table 2. We here compare the outcomes of the naive and myopic strategies with the optimal using DP. The myopic (nearsighted) strategy relies on forward selection alone, as opposed to the forward-backward approach of DP. Using cost/revenue utility, the myopic strategy starts from the most lucrative prospect 3A. If this variable is dry, we update the network and find out that all the conditional DVs are negative. In particular $P(4A = oil|3A = dry) = 0.975$, and this

ensures that prospect 4A is no longer attractive. If 3A is oil, the success probabilities in most nodes increase significantly. In fact, six of the seven remaining DVs are positive. The myopic approach goes for the largest of all DVs, and selects 4A as the next candidate. If 3A is oil and 4A is dry, we still have one positive DV. Not surprisingly, this is the prospect *above* 3A in the graph, and we go for prospect 2A. If both 3A and 4A are oil, we go again for the most lucrative prospect which is 5B.

The optimal DP solution defines values $v_i(\boldsymbol{\omega}_0)$ in (3) are as follows: $[3352, 3952, 3595, 3427, 3852, 3926, 3443, 3738]$ for $[1A, 2A, 3A, 4A, 4B, 5A, 5B, 5C]$. Note that all these values are much bigger than the naive value of the field, which is natural since the correlation in the graph is high. The first selected prospect is then 2A, which has an intrinsic value close to 0, but a large influence on the neighboring nodes. If 2A is dry, we focus on another area (prospect 5A). If 2A is oil, we remain relatively close (3A). For the second stage, in the event of 2A dry: If 5A is dry, the network has been entirely killed, and we stop observing. If 5A is oil, we remain in the same area (5B). The third stage is shown in Table 2.

For the entropy-based design, we again compare myopic with a full DP based strategy. For myopic the first node selected is either 5B or 5C, since their contribution to the reduction of the entropy is highest (see Table 1). No matter if segment 5B is found dry or oil, we move away from the 5-nodes, since most of the uncertainty in that part of the graph has been resolved.

Using DP, the first selected node is $1A$. This node has a balanced prior probability and a high impact on the probability structure in the network. In fact, the entropy values $v_i(\boldsymbol{\omega}_0)$ are now as follows: $[0.8534, 0.8487, 0.8066, 0.7850, 0.7713, 0.8520, 0.8353, 0.8353]$. Nodes 4A and 4B, which are characterised by prior probabilities far from 0.5, get the lowest initial entropy reduction. Node 4A is nevertheless selected when 1A is dry and 5A is oil (see Table 2). In this situation, when the left and right part of the network has been explored, 4A is the one with the highest marginal uncertainty, $p(4A = oil|1A = dry, 5A = oil) = 0.445$. The price $P = P_i$ is set relatively low, and under the entropy criterion we keep observing no matter the outcomes of the first two nodes.

Strategy	Naive M	Myopic M	Sequential M	Myopic E	Sequential E
$i_{(1)}$	3A	3A	2A	5B	1A
$i_{(2)}\|x_{i_{(1)}} = dry$	4A	Q	5A	2A	5A
$i_{(2)}\|x_{i_{(1)}} = oil$	4A	4A	3A	4A	2A
$i_{(3)}\|x_{i_{(1)}} = dry, x_{i_{(2)}} = dry$	Q	Q	Q	1A	2A
$i_{(3)}\|x_{i_{(1)}} = dry, x_{i_{(2)}} = oil$	Q	Q	5B	1A	4A
$i_{(3)}\|x_{i_{(1)}} = oil, x_{i_{(2)}} = dry$	Q	2A	5A	2A	5C
$i_{(3)}\|x_{i_{(1)}} = oil, x_{i_{(2)}} = oil$	Q	5A	4A	1A	5B

Table 2: Results of sequential design for the motivating example. Utility is monetary based (M) and entropy based (E). Here, $i_{(1)}$, $i_{(2)}$ and $i_{(3)}$ are the first, second and third nodes selected. Q means to *quit* the strategy.

3 Clustering strategies for large graphs

For large graphs the number of possible scenarios to evaluate exceeds what is computationally tractable. The objective functions in (3) or (4) must then be approximated in some way. Brown and Smith (2012) use *clusters* to overcome the computational limitations and to get an upper

bound for the expected utility. We now apply this method to build sequences. We study different complexity levels in the sequential Bayes update of the probability structure.

3.1 Sequential strategies based on clustering

The idea is to partition a large graph in smaller subgraphs, which can be computed efficiently. Let C^d, $d = 1, \ldots, L$ be disjoint nodes of the entire node set $\mathcal{N}$, i.e. $C^d \cap C^e = \emptyset$, and $\cup_{d=1}^{L} C^d = \mathcal{N}$. We denote by $\mathbf{x}_{C^d}$ the random variables in cluster C^d, and $\boldsymbol{\omega}_{C^d}$ the cluster specific evidence. The number of nodes in cluster C^d will be in the order of one to around ten. The approximations we present here improve as the cluster sizes grow, with a large increase in computational cost. As an example of the increase in computing time, consider a situation with binary outcomes $k = 2$. The computing time for evaluating a size 2 cluster is about 0.007 seconds, for 5 nodes we have 0.37 sec, and for nine nodes it is 50 seconds.

To construct an approximate sequential design, we suggest to rank the clusters and select the optimal node within the best cluster. The ranking is based on DP within clusters, given the current information. Once we collect data in a cluster, we update the probabilities, use DP again, and get a new ranking. This provides the basis for the selection at the next stage of the sequential design.

It is important here to introduce the Gittins index (GI), see Gittins (1979) and Whittle (1980). We consider the cluster-wise GIs in the spirit of Brown and Smith (2012). First, consider a variation of (3), with a generic retirement value M instead of 0 in the decision rule. Moreover, assume that this DP equation is set up for each cluster, given the current evidence. We have expected value for cluster d given by:

$$v^d(\boldsymbol{\omega}, M) = \max_{i \in C^d} \left\{ \sum_{j=1}^{k} p(x_i = j) \left[r_i^j + \delta \max_{s \in C^d / \{i\}} \left\{ \sum_{l=1}^{k} p(x_s = l | x_i = j)(r_s^l + \ldots), M \right\} \right], M \right\}. \quad (5)$$

Now, when the computation is restricted to cluster C^d, the GI is $M_{C^d}(\boldsymbol{\omega})$, defined as the smallest retirement value M such that $v^d(\boldsymbol{\omega}, M) = M$. This is the value which makes the decision maker indifferent between retiring and continuing the sequential strategy. Below, we will discuss various levels of conditioning on the generic evidence $\boldsymbol{\omega}$ in (5).

Brown and Smith (2012) derived some important properties for the value function $v^d(\boldsymbol{\omega}, M)$, for any evidence $\boldsymbol{\omega}$. Figure 3 illustrates the value functions. Here, we plot $v^d(\boldsymbol{\omega}_{C^d}, M) - M$ for some clusters related to an example below, for fixed evidence. The GI corresponding to each cluster is the crossing point with the first axis. Note that the ordering of the clusters is not monotone in M, indicating the changes in decision paths within the cluster. The cluster-wise GIs determine the cluster to be selected at the current stage of the sequential design. We find the cluster with the largest GI by gradually reducing the M in conjunction with DP for the clusters. Since the value function is piecewise linear, solving $v^d(\boldsymbol{\omega}_{C^d}, M) - M = 0$ for fixed M, is relatively fast. One can also use theory from Markov decision processes to solve the DP as a linear programming problem, see Chen and Katehakis (1986) and Brown and Smith (2012).

To study the policies induced by this cluster strategy, we suggest to generate realizations of the selected nodes. This entails running hypothetical scenarios where we sequentially observe (i.e. sample the outcomes) of the chosen nodes, update the probabilities, and then proceed to the next stage. At subsequent stages we may move between clusters or stay in the same cluster. The chosen cluster often depends on the outcomes at the previously selected nodes.

Note that the updating step can be quite time-consuming. In principle, the evidence vector $\boldsymbol{\omega}$ in (5) is the full observation sequence until this stage, in all clusters, not only in cluster d. This means that all cluster probability models must be updated when we acquire new data at a node. A faster,

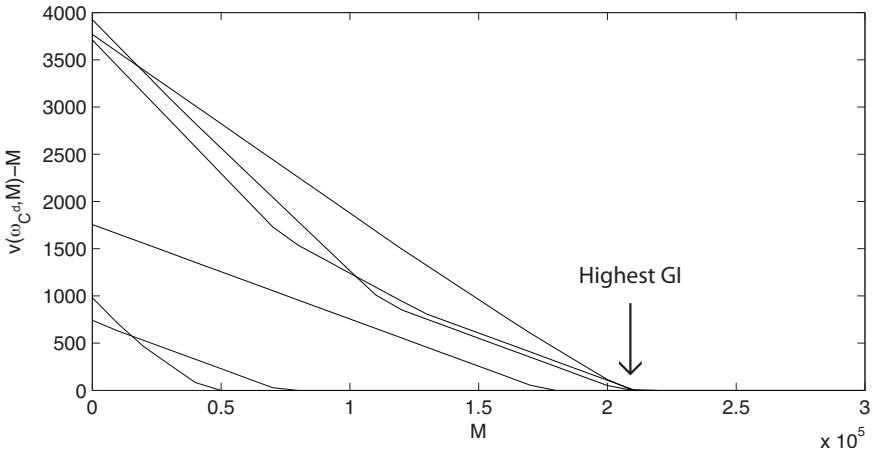

Figure 3: Illustration of GIs: Values of $v(\boldsymbol{\omega}_{C^d}, M) - M$ for six clusters, as a function of retirement value M. The GI is the crossing point with the first axis. The cluster with highest GI is selected in the sequential design.

but more approximate strategy is to update only the cluster where the current data is collected. This means that just one of the GIs changes at each stage. We implement both of these methods for updating the probability distribution, given the sequential observations. More specifically, we have:

- **Multiple clusters update (MCU)**: We rank the cluster according to their GIs. DP in the best cluster gives the first node. We update the probability model for *all* the clusters, given the observation (sample outcome) in the selected node. All cluster GIs are also modified based on the updated probabilities. Then, we choose the best cluster at the second stage using these GIs. We proceed until all the nodes have been observed or there are no more GIs greater than 0.

- **Single cluster update (SCU)**: We rank the clusters on the basis of their GIs. We start from the cluster with the biggest index, and select a node according to a DP strategy within the cluster. We update the joint distribution just for that cluster, given the observation in the first node. The probability model for the other clusters are not updated, and the GIs for other clusters then stays the same. Then, we choose the cluster with the highest GI among the updated cluster and the other initial cluster values. The cluster with highest GI is selected, and the best node based on DP in the cluster is chosen. We update this cluster only, given all data acquired so far. We continue until all the nodes have been observed, or until there are no more GIs greater than 0.

The MCU method is of course more consistent since we update the entire probability structure, and all GIs, whenever new information is available. The drawback is the computational

cost required to recompute the joint probability distribution and apply the DP strategy in every cluster after each observation. The SCU method is faster since only one cluster GI needs to be updated at each stage. However, the sequential design from SCU could suffer lack of accuracy. Pseudo-algorithms constructing sequences over Monte Carlo samples (observations) are described in Algorithm 1 and 2.

Algorithm 1 Evaluating a Single Cluster Update (SCU) strategy

$\boldsymbol{\omega} = [-, -, \ldots, -]$ # Dynamic programming outcome vector
$seq = [\,]$ # Best sequence vector
Sample $\mathbf{t} \sim p(\mathbf{x})$ # Current sample
for Clusters $d = 1 : L$ **do**
  $[v_{C^d}, s_{C^d}] = v(\boldsymbol{\omega}_{C^d})$ # Initial cluster-based DP values
  $GI_{C^d} = M : \; v(\boldsymbol{\omega}_{C^d}) - M = 0$ # Initial GIs
end for
while $\exists\, d : v_{C^d} > 0$ **do**
  $C^* = \arg\max_d \{GI_{C^d}\}$ # Best cluster
  $seq = [seq \; s_{C^*}]$ # Best node in cluster C^*
  $\omega_{s_{C^*}} = t_{s_{C^*}}$ # Set sampled outcome $t_{s_{C^*}}$ at selected node s_{C^*}
  $[v_{C^*}, s_{C^*}] = v(\boldsymbol{\omega}_{C^*, s_{C^*}}^{\,t_{C^*}})$ # Updated cluster-based DP value for cluster C^*
  $GI_{C^*} = M : \; v(\boldsymbol{\omega}_{C^*}) - M = 0$ # Updated GI for cluster C^*, according
  # to the new local conditional distribution.
end while

Algorithm 2 Evaluating a Multiple Clusters Update (MCU) strategy

$\boldsymbol{\omega} = [-, -, \ldots, -]$ # Dynamic programming outcome vector
$seq = [\,]$ # Best sequence vector
Sample $\mathbf{t} \sim p(\mathbf{x})$ # Current sample
for Clusters $d = 1 : L$ **do**
  $[v_{C^d}, s_{C^d}] = v(\boldsymbol{\omega}_{C^d})$ # Initial cluster-based DP values
  $GI_{C^d} = M : \; v(\boldsymbol{\omega}_{C^d}) - M = 0$ # Initial GIs
end for
while $\exists\, d : v_{C^d} > 0$ **do**
  $C^* = \arg\max_d \{GI_{C^d}\}$ # Best cluster
  $seq = [seq \; s_{C^*}]$ # Best node in cluster C^*
  $\omega_{s_{C^*}} = t_{s_{C^*}}$ # Set sampled outcome $t_{s_{C^*}}$ at selected node s_{C^*}
  for Clusters $d = 1 : L$ **do**
    $[v_{C^d}, s_{C^d}] = v(\boldsymbol{\omega}_{C^d, s_{C^*}}^{\,t_{C^*}})$ # Updated cluster-based DP value for cluster C^d,
    # according to the new conditional distribution.
    $GI_{C^d} = M : \; v(\boldsymbol{\omega}_{C^d}) - M = 0$ # Updated GI for cluster C^d
  end for
end while

3.2 Computing independent and sequential lower bounds and an upper bound

Associated with a cluster-based sequential design we can approximate the expected utility value $v(\boldsymbol{\omega}_0)$. Of course, the clustering strategy gives a sub-optimal value compared to the full DP solution,

but the optimal one is not tractable for large graphs. A useful aspect of the clustering approach is that we can get upper bounds for the value $v(\boldsymbol{\omega}_0)$ by using clairvoyant information.

Let us first discuss various ways of approximating $v(\boldsymbol{\omega}_0)$. The Monte Carlo strategies in Algorithm 1 and 2 provide a sampling-based approach for estimating this value. Here, each Monte Carlo sample constructs a design sequence which depends on the outcome at the selected nodes. We sum the $t_{s_{C^*}}$ selected at every step of the `while` cycle in Algorithm 1 and 2, possibly with discounting. Finally, these output values are averaged over B Monte Carlo runs. The estimates will differ between MCU and SCU, since the multiple updating scheme gives better sequences on average. A challenge with this Monte Carlo sampling approach is a large associated Monte Carlo error for moderate B.

Simpler approximations exist if we disregard the discounting. For instance, we get a lower bound on the intial value $v(\boldsymbol{\omega}_0)$ through an independent evaluation on each of the clusters: Let $v(\boldsymbol{\omega}_{C^d})$ be the DP value computed based on the evidence vector $\boldsymbol{\omega}$ restricted to cluster d as follows:

$$v(\boldsymbol{\omega}_{0,C^d}) = \max_{i \in C^d} \left\{ \sum_{j=1}^{k} p(x_i = j) \left[r_i^j + \delta \max_{s \in C^d / i} \left\{ \sum_{l=1}^{k} p(x_s = l | x_i = j)(r_s^l + \ldots), 0 \right\} \right], 0 \right\}. \qquad (6)$$

A lower (independent) bound for the expected utility is defined by the sum of the marginal values for all the clusters:

$$v_{LB(1)}(\boldsymbol{\omega}_0) = \sum_{d=1}^{L} v(\boldsymbol{\omega}_{0,C^d}).$$

Clearly, $v_{LB(1)}(\boldsymbol{\omega}_0) \leq v(\boldsymbol{\omega}_0)$, since this cluster-by-cluster approach ignores the dependence between clusters. However, this procedure requires no simulations, and if the clusters are chosen well, the bound can be reasonable.

This lower bound defined via (6) can be improved by sequential cluster selection. Assume we start by evaluating the cluster with the highest GI. Its value is $v(\boldsymbol{\omega}_{0,C^d})$. We next generate an outcome for this cluster $\mathbf{t}_d$, and use DP restricted to this cluster, plugging in the sampled data at the selected nodes. Based on this we update the probability model at the remaining $L-1$ clusters, and choose the next cluster with highest GI, say C^e, and so on. This sequential cluster average value defines an improved lower bound. Over B Monte Carlo samples we have

$$v_{LB(2)}(\boldsymbol{\omega}) = \frac{1}{B} \sum_{b=1}^{B} \left[\sum_{d=1}^{L} v(\boldsymbol{\omega}_{C^d} | \mathbf{t}_b^{C^{e<d}}) \right],$$

where the conditioning is the empty set for $d = 1$, and $\mathbf{t}_b^{C^{e<d}}$ is the b^{th} sample restricted to all clusters considered previously. Because of the imposed learning, we have $v_{LB(1)}(\boldsymbol{\omega}_0) \leq v_{LB(2)}(\boldsymbol{\omega}_0) \leq v(\boldsymbol{\omega}_0)$. The quality of this sequential strategy depends on the choice of clusters and on the Monte Carlo sample size B.

We next consider the construction of an upper bound. This is based on clairvoyant information in the sequential strategy. This means that we know the outcome of all other nodes, and use this when making decisions at the current stage. Since we are using information that is not really available in practice, we get an upper bound for the initial value $v(\boldsymbol{\omega}_0)$. This works as a benchmark for sequential strategies. Together with the various lower bounds, we can squeeze the initial value.

The Monte Carlo strategies in Algorithm 1 and 2 can be extended to provide a sampling-based approach for the upper bound. Now, at each stage, the GIs are computed by DP within-cluster, using the updated probability model, given the cluster evidence available at the current stage (if any), and all sample values outside the cluster. If we again disregard discounting, we can solve the

clairvoyant bound separately for each cluster. In this case we compute the value of a cluster, given all observations (samples) outside the cluster. This calculation requires the full conditional for all clusters, given the sampled outcomes in other clusters, but there is no computation of GIs. The upper bound of the initial value is in this case:

$$v_{UB}(\boldsymbol{\omega}) = \frac{1}{T}\sum_{b=1}^{B}\left[\sum_{d=1}^{L} v(\boldsymbol{\omega}_{C^d}|\mathbf{t}_b^{C^{e\neq d}})\right]$$

where $\mathbf{t}_b^{C^{e\neq d}}$ is the b^{th} sample restricted to clusters different from C^d. In summary, we get

$$v_{LB(1)}(\boldsymbol{\omega}_0) \leq v_{LB(2)}(\boldsymbol{\omega}_0) \leq v(\boldsymbol{\omega}_0) \leq v_{UB}(\boldsymbol{\omega}_0).$$

4 Synthetic examples

We first study small BNs and MRFs to compare various cluster configurations. The number of nodes is at most 12, and we manage to compare the clustering sequences and values with the optimal solution obtained by full DP.

4.1 Small BN : Entropy utility

The entropy reduction is relevant in many applications, see e.g. Marcot et al. (2001) and Aalders et al. (2011). When the BNs get large, and sequential strategies are requested, the current approach should be interesting.

We run (4) on two small BNs, shown in Figure 4. The two BNs are small clusters of a bigger network, connected through a Common Parent (CP) node. Both BNs have five nodes which can be selected. In the network on the left the structure is made by a common node and four children, while the network on the right has two chains departing from a common top node. Each node is binary, with two states denoted A and B.

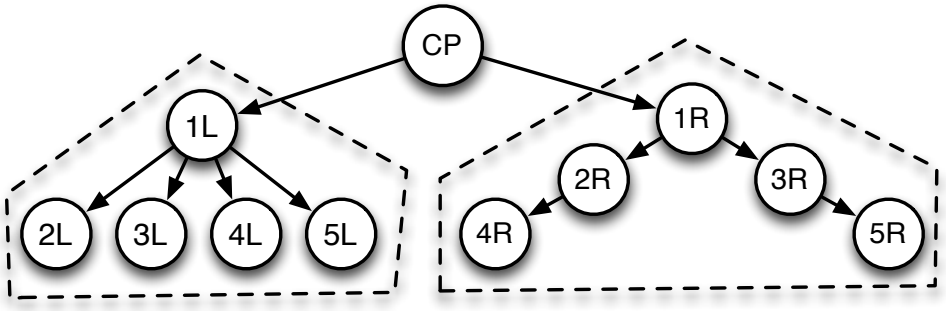

Figure 4: Simple BNs used for testing the entropy criterion, connected through a

Let us start with the left network. The top node has a symmetrical prior, i.e. 0.5 probability of state A. Nodes 2 and 3 have a CPT with propagation of information just through state A, while nodes 4 and 5 have perfectly balanced CPT, as shown in Table 3

We set $\delta = 1$. The original entropy of the network in configuration $\boldsymbol{\omega}_0 = \{-,-,-,-,-\}$ is 2.3615. We intuitively expect that the reduction in entropy is higher if we observe the parent node

$x_2, x_3 \setminus x_1$	A	B
A	0.9	0.1
B	0.5	0.5

$x_4, x_5 \setminus x_1$	A	B
A	0.9	0.1
B	0.1	0.9

Table 3: CPT for Multi Level Network, from level nodes to children nodes

1. However, due to symmetry we get:

$$\sum_{x_1} p(x_1) \left\{ \sum_{\mathbf{x}} -p(\mathbf{x}|x_1) log(p(\mathbf{x}|x_1)) \right\} = \sum_{x_4} p(x_4) \left\{ \sum_{\mathbf{x}} -p(\mathbf{x}|x_4) log(p(\mathbf{x}|x_4)) \right\} = 1.6684,$$

and the same holds for node 5. The entropy is substantially reduced if we observe state A in either of these three nodes. Then, a single configuration, $\{A, A, A, A, A\}$ gets 0.65 conditional probability. If we observe B in 1 or 4/5, the result is having four configurations equally likely, $\{B, A, A, B, B\}$, $\{B, A, B, B, B\}$, $\{B, B, A, B, B\}$ or $\{B, B, B, B, B\}$, each with a little more than 0.20 conditional probability. Node 2 and 3 are not that informative. The overall effect is that an observation in 1 or 4/5 produces an average decrease in entropy of 0.6931. In comparison, the reduction brought by an observation in nodes 2 or 3 is 0.6109.

The question now is, are the results that intuitive when a full DP strategy is used, i.e. when we have the possibility to keep observing until the entropy reduction is smaller than fixed cost P? The results (final values for all the nodes) are reported in Table 4, left.

Left network	1	2/3	4/5
DP, P=0.2	1.3615	1.4828	1.4828
DP, P=0.5	0.3863	0.3803	0.4234
DP, P=0.65	0.0863	< 0	0.0823
Myo, P=0.2	0.4931	0.4109	0.4931
Myo, P=0.5	0.1931	0.1109	0.1931
Myo, P=0.65	0.0431	< 0	0.0431

Right network	1	2/3	4/5
DP, P=0.2	1.2135	1.1607	1.2135
DP, P=0.5	0.2055	0.3139	0.4139
DP, P=0.65	0.0431	0.0431	0.0431
Myo, P=0.2	0.4931	0.4931	0.4931
Myo, P=0.5	0.1931	0.1931	0.1931
Myo, P=0.65	0.0431	0.0431	0.0431

Table 4: Final values of the DP and Myopic strategies applied to the networks in Figure 4, for different prices P of experiment, and for $\delta = 1$.

We see some surprising results: in the myopic case (Myo), the sequence starts by the reduction in entropy brought by the first node, and therefore it is not surprising that nodes 1 and 4/5 emerge as winners, no matter the price $P_i = P$ of data collection. If the price is higher than 0.6931, no node is profitable, because this is the maximum reduction in entropy that can be achieved with a single-node observation. For DP strategies the final values are higher than with myopic. The results are harder to interpret for DP: if the price is very high (0.65), we might have to stop after a single observation. That is why node 1 is selected as the best choice. Even though the average reduction for node 1 and nodes 4/5 is the same, the marginal entropy of $\{A, -, -, -, -\}$ is smaller than $\{-, -, -, A-\}$. Therefore an observation A in node 1 could be sufficient, but an observation A in node 4/5 may not. This reflects in the different final values. When the cost is medium (0.5), node 4/5 are selected first. At this price level we choose to observe two or three nodes (depending on their outcome). When the cost is very small (0.2), it is convenient to keep observing up to end, and this makes the values for nodes 2/3 and 4/5 identical.

Results for the network on the right in Figure 4 are shown in Table 4, right. Here, all the nodes have symmetrical CPT, and the marginal reduction in entropy equals 0.6931 for all five nodes.

Because of this symmetry the results for the three node sets are equal in the myopic strategy. The same holds for DP with high price level (0.65), since just one observation is allowed.With lower prices, it is optimal to start far from the center. Often, two or more observations are allowed, and the reduction in entropy is higher by observing nodes 4 and 5 than observing node 1 and any other node.

When considering the full network, we regard the two subnetworks as two different clusters, and set $P = 0.5$. The left cluster has the highest GI and is selected first. This should not be surprising, since the left network represents a less informed case with a total entropy of 2.36, while the the second cluster has a total entropy 1.99. Within the left cluster, one of either node 4 or 5 are selected. After observing either of these two nodes, the right cluster gets the highest GI, and the decision maker move there to find a node that reduces the entropy. The suggested ones are node 4 or 5. The choice does in this case not depend on the outcome of the first selected node, but this is in general not true. In particular, as we have seen before, if the outcome is A, the entropy in the first cluster is drastically reduced and the indication of moving to the second cluster is strong. If the outcome is B, the entropy is only slightly reduced, just enough to make the right cluster more valuable in terms of GI. The sequential strategy (SCU or MCU) suggests to keep alternating nodes between left and right cluster.

This small example is synthetic, but similar situations may arise in real problems. We can imagine two sub clusters representing different areas that share little information, and we have to place monitor stations in order to maximise the entropy reduction of our data. Our proposed approach uses statistical updating to guide the selection of where to place the first stations, the second station, given the first, and so on.

4.2 Small BN: revenues/ cost

We focus on two small BNs (Figure 5) with 12 correlated prospects. The small dimension allows exact solutions. We analyse sequences and show how different network structures influence the quality of the bounds. We further compare these bounds with the ones produced by the approximated sequential strategies presented in Martinelli et al. (2011a).

Figure 5 (left) shows 12 prospects mutually correlated through a single common parent. For a similar use of common parent networks in oil and gas exploration contexts, see Martinelli et al. (2012). Table 5 shows the marginal probabilities and the intrinsic values (IV)s for all 12 prospects. We consider an oil exploration setting, with two possible outcomes, *oil* and *dry*. We have few

prospect i	1	2	3	4	5	6	7	8	9	10	11	12
$p(x_i = dry)$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
$p(x_i = oil)$	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Intrinsic Value	-108	375	-657	-711	360	-184	-172	2417	-642	815	1088	-949

Table 5: Marginal probabilities and Intrinsic Values for the 12 prospects of the case studies in Section 4.2

prospects with positive IV, the most prominent being prospects 8, 10 and 11. The network CPTs (Table 6) impose both positive and negative correlations between the prospects. For example, discovering oil in prospect 8 boosts the probability of a discovery in prospect 11, but lowers the probability of finding oil in prospects 10 and 7.

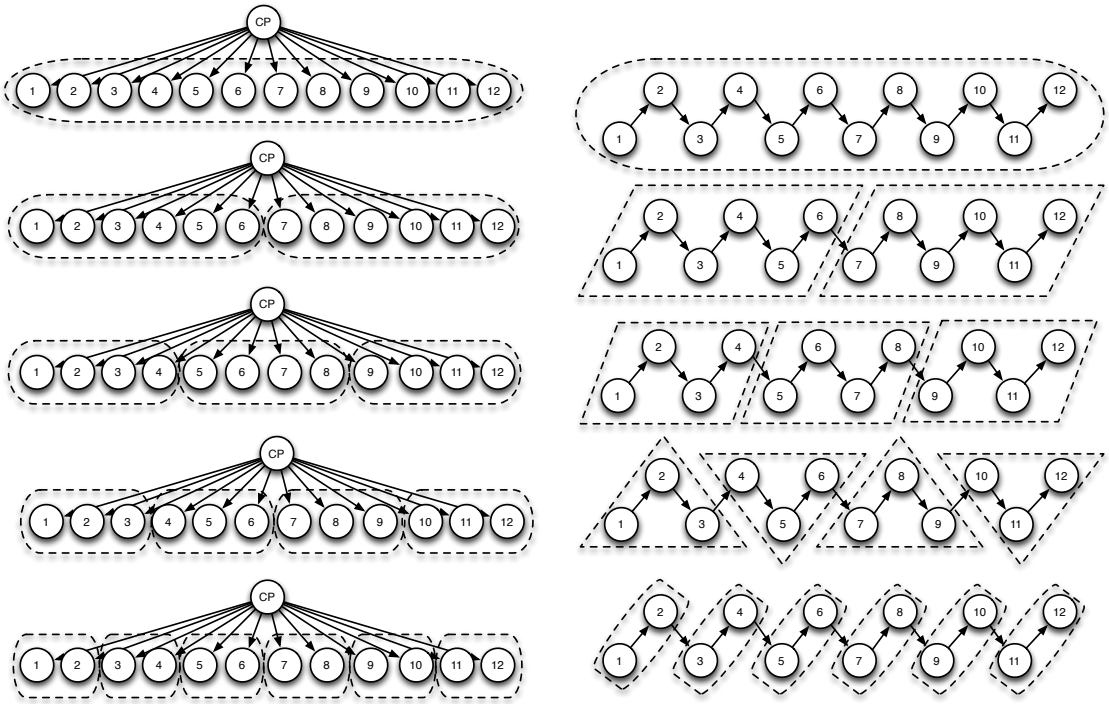

Figure 5: Small BN case studies used in Section 4.2.

P1, P4, P7, P10 / CP	0	1
dry	0.2	0.8
oil	0.8	0.2

P2, P5, P8, P11 / CP	0	1
dry	0.8	0.2
oil	0.2	0.8

P3, P6, P9, P12 / CP	0	1
dry	0.6	0.4
oil	0.4	0.6

Table 6: Conditional Probability Tables for the children nodes of the case study presented in Section 4.2, shown in Figure 5, on the left

In Figure 5 (right) the dependence structure is imposed through a Markov chain. The resulting marginals and the IV are the same as in the previous structure (0.5 chance of oil in each of the prospects), but now the CPTs link directly the prospects with each others and not through a common parent. Thus, there is a predetermined order that guides the clustering, and we here aim to study this structuring effect on the resulting cluster strategies.

We study the sequential SCU strategies under different clustering configurations. We use discounting $\delta = 0.99$. Results for the left network are summarized in Table 7. For the 2-clusters approach (second column), the first selected node is prospect 8, characterized by the highest IV. If prospect 8 is dry, we choose prospect 10, which is negatively correlated with prospect 8. If prospect 8 is oil, we move to prospect 11, which is positively correlated with prospect 8. And so on for the third best choice. In general we observe less flexibility in strategies characterised by many clusters (left columns).It is interesting to notice that the strategy with 12 clusters using SCU, coincides with the naive strategy, and the first three selected prospects are 8, 11 and 10, no matter their outcome.

# Clusters	1	2	3	4	6	12
Cluster size	12	6	4	3	2	1
$i_{(1)}$	8	8	8	8	8	8
$i_{(2)}\|x_{i_{(1)}} = dry$	10	10	10	10	11	11
$i_{(2)}\|x_{i_{(1)}} = oil$	2	11	10	10	11	11
$i_{(3)}\|x_{i_{(1)}} = dry, x_{i_{(2)}} = dry$	5	11	11	11	7	10
$i_{(3)}\|x_{i_{(1)}} = dry, x_{i_{(2)}} = oil$	1	7	7	7	7	10
$i_{(3)}\|x_{i_{(1)}} = oil, x_{i_{(2)}} = dry$	5	10	11	11	10	10
$i_{(3)}\|x_{i_{(1)}} = oil, x_{i_{(2)}} = oil$	11	5	11	11	10	10

Table 7: Results of the sequential exploration program SCU for the simple BN example described in Section 4.2, for strategies with different cluster size. $i_{(1)}$, $i_{(2)}$ and $i_{(3)}$ are respectively the first, the second and the third best prospect selected. Q means *quit* (the strategy).

The final values for left and right networks are shown in Table 8. Here we show results over 1000 Monte Carlo samples using either the MCU or SCU approach. This means that for each sample and for each strategy we simulate outcomes of the drilling campaign, collecting the revenues for the wells whose outcome turned out to be positive, the costs for the sites whose outcome was negative, and stopping when the strategy would not support more exploration.

Naturally, the final values of MCU strategies always improve the final values of SCU strategies. The final value with the SCU monotonically increases with cluster size, reaching its maximum when all the sites are gathered in a single cluster. With the MCU approach, the quality of the results does not decrease so much with smaller cluster size. Note that the SCU and MCU are much closer to each other for the right Markov chain network. This reflects the behaviour of SCU and MCU sequential strategies. Since there is a natural ordering in the clusters, the learning within the cluster is higher in the Markov chain network. This means that SCU strategies with large cluster size (six or four nodes per cluster) perform much better than for the more unstructured common parent network.

We next set $\delta = 1$ and compare the sequential cluster strategies with the lower and upper bounds. Figure 6 shows the result of left and right network, plotted as a function of cluster size. The 2-clusters configuration reveals already a high efficiency, with a computational time that is orders of magnitudes less than the single cluster configuration. Because of Monte Carlo variability the sequential bounds for the 2-clusters approach look tighter than the bounds for the 3-clusters approach (left display). The Markov chain network (right) has narrower gaps in the bounds. In the common parent structure we have conditional independence between children only through the common parent, which is not possible to observe directly. This makes the learning very hard. With the Markov chain structure, on the other hand, we have conditional independence between clusters, given a separating cluster, and this makes the clustering approach more efficient. Still, in the left display the sequential value (LB2) remains quite close to the optimal value, even for relatively small clusters. The independent value (LB1) increases with cluster dimension, but it is never as good as the sequential bound. Correspondingly, the clairvoyant bounds (UB) slightly decreases as the clusters become larger. In general, an increase in the cluster size does not seem to have a strong effect. The gap between the sequential and the upper bound is 1321 for clusters of size 3, and 1421 for clusters of size 2. For the Markov chain structure the gap between the sequential and the upper bound is 149 for clusters of size 3, and 371 for clusters of size 2.

We also compare these results with the ones obtained via Rolling-Horizon Look-Ahead (RHLA)

strategies presented in Martinelli et al. (2011a). These strategies do not use a clustering approach based on the BN structure. They instead approximate the DP value after a certain number n of steps with heuristics approximations, resulting in *Depth n* (Dpt n) RHLA strategies. For increasing depths, the value goes towards the correct value, because we are using the heuristics after a higher number of exact steps. We observe that for the common parent network (left) it is sufficient to compute a Dpt 6 strategy for reaching a value that is very close to the exact one. From a computational point of view, a Dpt 6 strategy costs as much as a clustering strategy with cluster size equal to 6, with a substantial improvement in the quality of the approximation (the gap between a Dpt 6 strategy and the exact one is less than 10 units). For the Markov Chain network (right), on the other hand, the comparison with the RHLA strategies shows that the quality of a clustering strategy with cluster size equal to four or six, performs as well as a RHLA strategy with corresponding depth, showing that in this case the approximation introduced by a clustering approach is reasonable and effective.

# Clusters	1	2	3	4	6	12	Case study
Cluster size	12	6	4	3	2	1	(Figure 5)
Average Value SCU	7295	6992	6779	6638	6398	5783	Left
Average Value MCU	7295	7153	7112	7112	7112	7112	Left
Average Value SCU	7962	7887	7876	7869	7159	4947	Right
Average Value MCU	7962	7887	7887	7887	7765	7762	Right

Table 8: Average values (discounted final values for a sample of size 200), for clusters of different size, for the two case studies presented in Section 4.2.

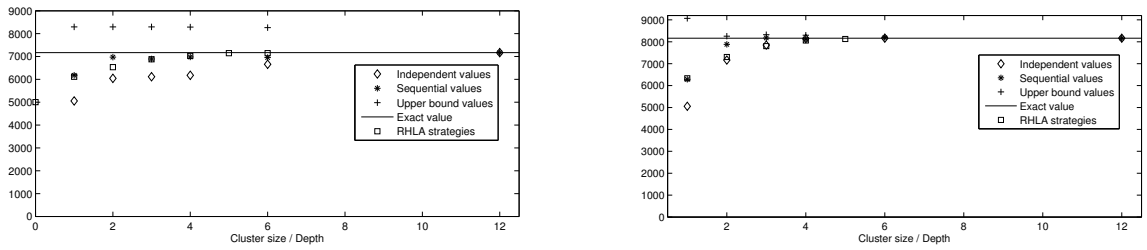

Figure 6: Independent LB, sequential LB, clairvoyant UB and exact values for the two examples shown in Section 4.2. We also show the comparison with RHLA strategies.

4.3 MRF : small cases

We next study MRFs. We use a lattice where each node represents a reservoir unit, i.e. a possible prospect. We are interested in finding the best drilling sequence and approximating the expected value of the whole field. Here, we test our methods on a small 3×4 lattice with 12 nodes corresponding to 12 potential prospects. We compare the exact DP solution (which is possible for such small lattices) with cluster approaches splitting the lattice in a number of sub-lattices. The MRF is an Ising field ($\beta = 1$) with $k = 2$ colours, representing oil and dry states. Revenues and costs are equal for all prospects and symmetrical ($+3$ and -3 units). The field is non-symmetrical, i.e. a priori marginal probabilities for oil and dry states follow a parabolic trend with a maximum in cell 6 (2nd row, 2nd column). As a direct consequence, the IVs are oscillating around 0, with positive

values just on the left part of the lattice. Marginal prior probabilities and IVs are shown in Figure 7. The nodes are numbered from left to right and from top to bottom, see Figure 8. We test the effects of different clusters size and shape. We propose six possible cluster combinations in Figure 8.

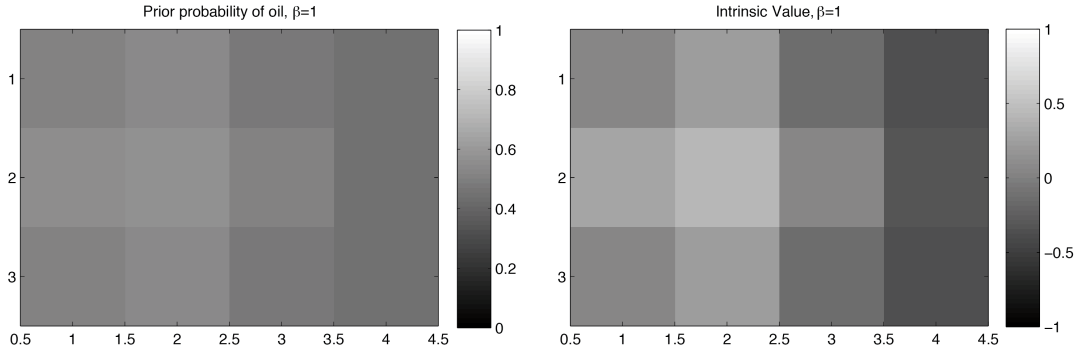

Figure 7: Marginal prior probabilities of oil and Intrinsic Values for the case study presented in Section 4.3

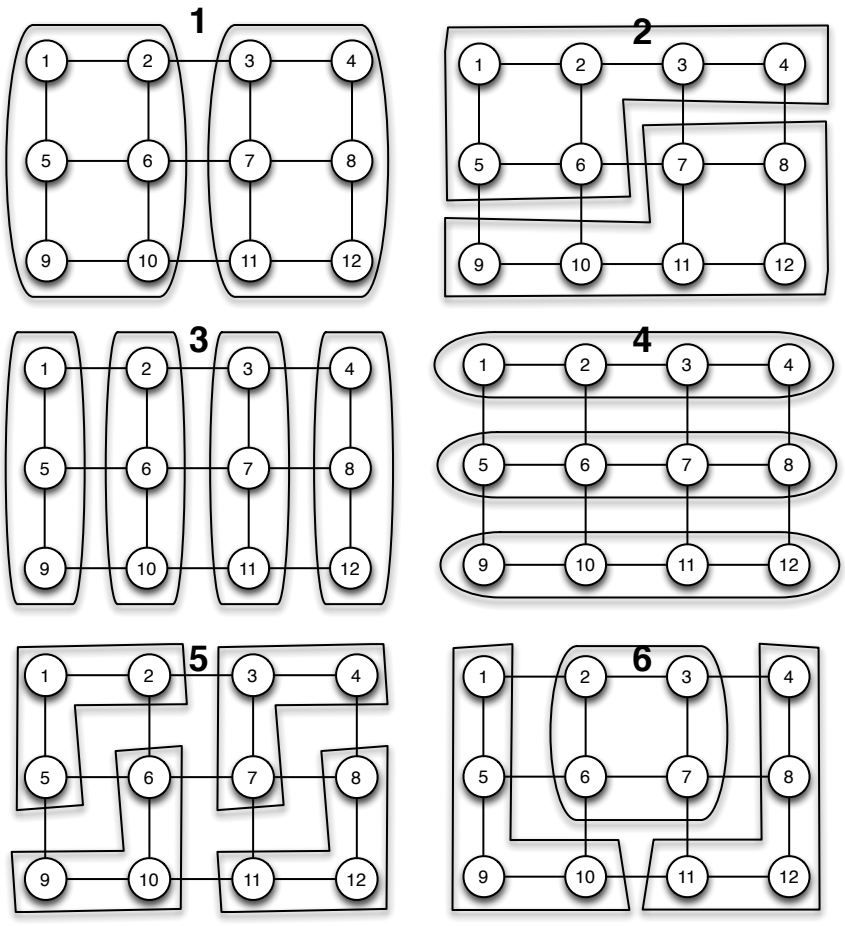

Figure 8: Six possible clusters' configurations for the MRF presented in Section 4.3

We analyse the best sequences computed with SCU (Table 9), setting $\delta = 0.99$. We notice that, given the high correlation present in the field, if the first node is found dry, there are few hopes to find any good spots. This is the reason for quitting (Q) the strategy for the 1-cluster scenario, when the first node (prospect 6) is dry. When more than one cluster is used, the strategy moves out from the cluster where the dry node has been found, but suggests to keep the exploration campaign alive. When oil is found in the first place, the suggestion is to keep drilling in the same cluster where the discovery has happened. The optimal choice is to drill prospect 7, which is crucial for exploring the right part of the field.

It is interesting to observe that configurations 3 and 4 are more rigid, in the sense that the third best choice does not depend on the outcome of the second, but just on the outcome of the first. This reflects the vertical and horizontal clusters present in configurations 3 and 4, that impose less learning in the MRF. One way to compare the strategies is to study the final values shown in the last row of Table 9. These are based on 100 Monte Carlo simulations. For the sake of comparison, the naive value of the field (sum of positive IVs) is 1.31. There is an evident reduction of value when the cluster number increases, as expected. There is also a less immediate, yet interesting and comforting, increase in value for more compact clusters. I.e. results for configuration 1 are better than results for configuration 2, and results for configuration 6 are better than those for configuration 4. The results of the exact DP are in this case much better than any of the clustering configurations. This appears to be a consequence of the decision of quitting right after the first dry node, without further losses.

Configuration	0	1	2	3	4	5	6
# Clusters	1	2	2	4	3	4	3
Cluster size	12	6	6	3	4	3	4
$i_{(1)}$	6	6	6	6	6	6	6
$i_{(2)}\|x_{i_{(1)}} = dry$	Q	7	10	5	10	5	5
$i_{(2)}\|x_{i_{(1)}} = oil$	7	2	5	10	5	10	2
$i_{(3)}\|x_{i_{(1)}} = dry, x_{i_{(2)}} = dry$	Q	Q	Q	2	2	7	11
$i_{(3)}\|x_{i_{(1)}} = dry, x_{i_{(2)}} = oil$	Q	8	9	2	2	7	11
$i_{(3)}\|x_{i_{(1)}} = oil, x_{i_{(2)}} = dry$	5	10	2	7	7	5	5
$i_{(3)}\|x_{i_{(1)}} = oil, x_{i_{(2)}} = oil$	2	5	2	7	7	9	7
Average Value	8.02	6.98	6.89	5.10	5.95	5.01	6.30

Table 9: Results of the sequential exploration program for the simple MRF example described in Section 4.3, for strategies with different cluster size and shape. $i_{(1)}$, $i_{(2)}$ and $i_{(3)}$ are respectively the first, the second and the third best node selected. Q means *quit* (the strategy).

5 Real examples

The next two examples are from the petroleum industry. Drilling decisions for large-scale prospects is the challenge in the first application, using a BN model to model the dependence between prospects. The second example is at a smaller geographical scale, where many reservoir units are represented using a MRF. In both situations the clustering strategies are useful for improved drilling campaigns, letting both exploration and exploitation play integral parts.

5.1 Large BN for prospects in the North Sea

The original motivation for this work comes from a large BN describing a geological feature: the migration paths of the Hydrocarbons (HC) expelled by the source rock in a field located in the North Sea. The network and its parameters were originally presented in Martinelli et al. (2011b), and extensively discussed for similar purposes of optimal exploration in Martinelli et al. (2011a) and Brown and Smith (2012). The graph is composed by 17 auxiliary nodes (kitchens, marked with K and prospects, marked with P in Figure 9), and 25 segment nodes at the bottom. The segments correspond to possible drilling locations. Each segment node can have $k = 3$ states (oil, gas or dry), and the conditional distributions are assigned to obey physical constraints.

Here, we consider two possible ways to divide the original network in clusters. The first one,

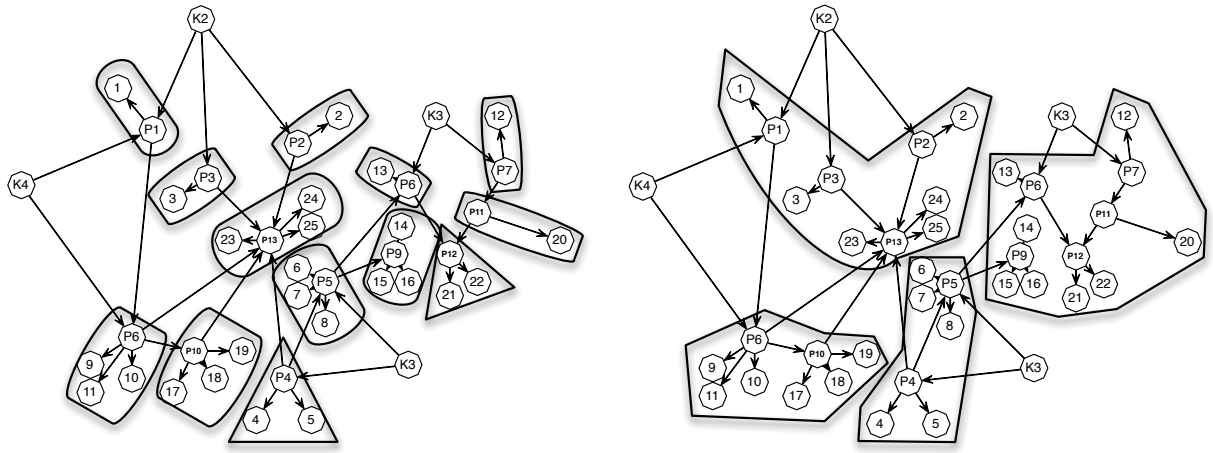

Figure 9: Bayesian Network describing the migration paths of the HC expelled from the source rock. The letter K marks the *kitchens*, i.e. places where the formation of HC has started, the letter P marks the *prospects*, large areas of possible accumulations, while the numbers mark the segments, corresponding to potential drilling locations.

with clusters of small dimension, is shown in Figure 9, left. The second one, with clusters of bigger dimension, is presented in Figure 9, right. Brown and Smith (2012) describe the effect of different cluster size on the bounds, and show that the gap is sufficiently small even for moderate-size clusters. This is not surprising, given that the learning is relatively local for this network. Similar comments can be found also in Martinelli et al. (2011a). Now we are interested in studying how the strategies described in Section 3.1, for both SCU and MCU, perform on this large network. The average results over 200 Monte Carlo samples are reported in Table 10.

	Small clusters		Big clusters	
	SCU	MCU	SCU	MCU
Estimated value	22637	23117	23981	24001
Average # nodes drilled	17.07	16.75	18.02	17.70
Average # nodes dry	3.23	3.05	3.10	3.00
Time per sample	15sec	25sec	30min	50min

Table 10: Sequential clustering strategies applied to the case presented in Section 5.1. Small clusters refer to the partition of Figure 9 on the left, while big clusters refer to the partition on the right

The expected revenues increase with cluster size, as expected, and increase when we update every cluster with the new information (MCU) and not just the cluster where we have collected the last piece of information (SCU). The difference between SCU and MCU is larger when the clusters are small and many, since SCU with small clusters may disregard an impact on a large part of the network. When the cluster size gets larger, we are loosing just peripheral information when using SCU instead of MCU.

Note that the average number of nodes drilled increases when going from small to big clusters, but the number of nodes drilled and found dry decreases, meaning that we are more accurate. What is more surprising is that when we move from SCU to MCU both the average number of nodes and the number of dry nodes decrease, but with an increase in the revenues. This means that we are *avoiding to drill* just the dry nodes, while we are keeping the good nodes. This is of course good news for a petroleum company, who can now likely increase their profits by applying these more sophisticated statistical updating schemes in their work processes. In this whole analysis we can not ignore the fact that applying MCU with big clusters is computationally more expensive.

The difference between SCU and MCU (Table 11) for the small-clusters partition does not appear in the first nor in the second choice, but just at the third choice. The equal second choice is due to the fact that cluster 6, that includes segment 18, the best segment, remains the one with highest GI even after segment 18 is selected, if the update is positive (oil or gas). Therefore it is selected as second best choice by both approaches. When segment node 18 is dry, we move to the cluster containing segment 8, which is far away in the graph, and whose GI is selected by both MCU and SCU approaches. For what concerns the third choice, we can, on the other side, see the difference between the two approaches: in the MCU approach, after leaving cluster 6 with a good outcome (at least one segment oil or gas), we move to a neighbouring cluster and we drill segment 9. In the SCU approach, where the cluster containing segment 9 has not received the positive information, we pick the second cluster with the a priori highest GI, moving far away towards segment 8. In the big-clusters partition the same hold, but now segment 9 and 18 are in the same cluster, and we can no longer notice differences between the two approaches, at least in the first 3 choices. Considering these sequences, and the large increase in computational time using large clusters, it seems that MCU with small clusters is a flexible and fast option.

	Small clusters		Big clusters	
	SCU	MCU	SCU	MCU
$i_{(1)}$	18	18	18	18
$i_{(2)}\|x_{i_{(1)}}$ =dry	8	8	8	8
$i_{(2)}\|x_{i_{(1)}}$ =oil or gas	19	19	19	19
$i_{(3)}\|x_{i_{(1)}}$ =dry, $x_{i_{(2)}}$ =dry	10	24	24	24
$i_{(3)}\|x_{i_{(1)}}$ =dry, $x_{i_{(2)}}$ =oil or gas	10	24	24	24
$i_{(3)}\|x_{i_{(1)}}$ =oil or gas , $x_{i_{(2)}}$ =dry	8	9	9	9
$i_{(3)}\|x_{i_{(1)}}$ =oil or gas, $x_{i_{(2)}}$ =oil or gas	8	9	9	9

Table 11: Results of the sequential exploration program for the large BN case study shown in Section 5.1, for single cluster update and multiple clusters update strategies with different cluster size. $i_{(1)}$, $i_{(2)}$ and $i_{(3)}$ are respectively the first, the second and the third best segment node selected.

5.2 MRF for reservoir units in a North Sea oil field

This MRF case study is from an oil reservoir in the North Sea. Bhattacharjya et al. (2010) use this example to evaluate static acquisition strategies for imperfect data (seismic data acquisition). Here, we consider the sequential drilling problem over the dependent reservoir units. We use a lattice representation of the field with 10×4 cells, i.e. 40 nodes. The model is a categorical first-order MRF as in (2). The MRF model has three colours, representing respectively *oil saturated sand* ($x_i = 1$), *brine saturated sand* ($x_i = 2$) and *shale* ($x_i = 3$). Only the oil state is lucrative. The other two states entail only cost at location i, but may of course provide valuable information about the outcomes at other nodes. The external field parameter $\alpha_i(x_i)$ is set from geological information and from existing seismic data, see Bhattacharjya et al. (2010).

As was done in Bhattacharjya et al. (2010), we assign a fixed cost $r_i^j = 2$ Million USD for drilling a dry well (state $j = 2$ or $j = 3$), while we have a potential revenue $r_i^j = 5$ million USD when finding an oil saturated sand (state $j = 1$). Before drilling we have the situation represented in Figure 10, top display.

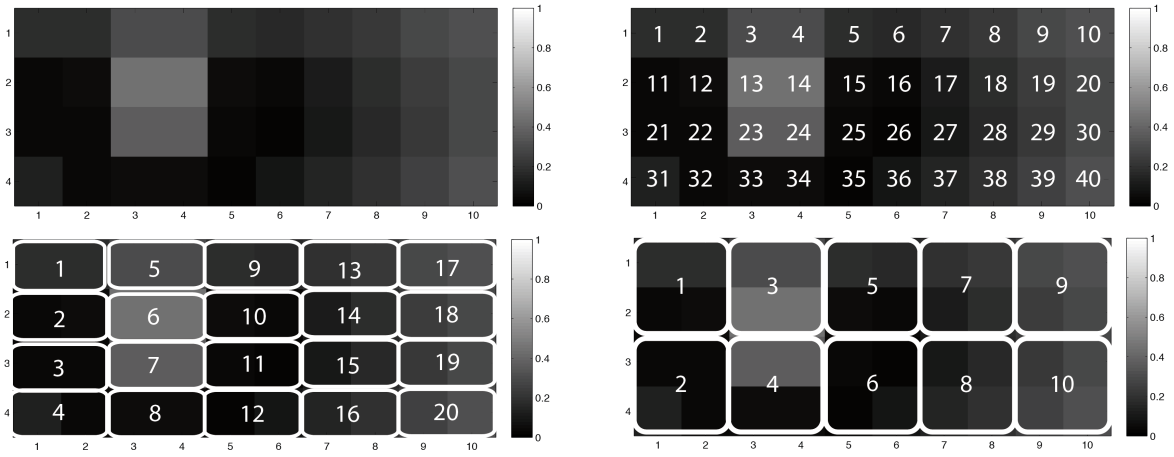

Figure 10: Top: marginal probability for state *oil* in the 10×4 $(1 - 40)$ reservoir units. Bottom left: 2-cells clusters. Bottom right: 4-cells clusters.

The 40-cells size of the MRF prevents us from running a full search. In Martinelli et al. (2011a) we have considered solutions based on an approximation with myopic/naive heuristics to the original DP procedure. Here we compare two possible clustering strategies with SCU approach, the first based on 20 very small 2-cells clusters, and the second based on 10 small clusters of size 2×2 (Figure 10, bottom display).Since the field is not homogeneous, we re-compute the joint cluster probabilities $p(\mathbf{x_{C(i)}})$ for all conditioning sets, and the computational time is large. Faster results could possibly be obtained by approximating the forward-backward algorithm used for computing $p(\mathbf{x})$, using the arguments presented in Tjelmeland and Austad (2012).

We also compare these results with those obtained with RHLA strategies of different depth presented in Martinelli et al. (2011a). Results in terms of final values and bounds are shown in Table 12.

We notice how in this case a clustering strategy with large clusters produce better results than the RHLA strategies until Dpt 2. Simple 2-cells clusters give much better results than the classical naive approach (sum of positive intrinsic values), and the result is further improved when using 4-cells clusters. It is worth noticing that the gap between sequential LB and Clairvoyant LB is

Cluster size	2-cells	4-cells	Naive	Dpt 1	Dpt 2
Independent LB	8.04	10.44	-	-	-
Sequential LB	9.12	13.71	-	-	-
RHLA LB	-	-	4.21	8.36	10.74
Clairvoyant UB	28.23	17.00	-	-	-

Table 12: Lower and upper bounds with clusters of different size and RHLA depth 1 and depth 2 final values, for the case study presented in Section 5.2. Parameters: $\beta = 1$.

already quite narrow with 4-cells clusters. The bounds are produced by averaging over 200 Monte Carlo samples.

	2-cells clusters	4-cells clusters	Dpt 1	Dpt 2
$i_{(1)}$	14	14	19	14
$i_{(2)}\|x_{i_{(1)}}$ =brine or shale	24	10	14	19
$i_{(2)}\|x_{i_{(1)}}$ =oil	13	13	14	19
$i_{(3)}\|x_{i_{(1)}}$ =brine or shale, $x_{i_{(2)}}$ =brine or shale	10	24	40	40
$i_{(3)}\|x_{i_{(1)}}$ =brine or shale, $x_{i_{(2)}}$ =oil	23	20	4	18
$i_{(3)}\|x_{i_{(1)}}$ =oil, $x_{i_{(2)}}$ =brine or shale	24	10	18	4
$i_{(3)}\|x_{i_{(1)}}$ =oil, $x_{i_{(2)}}$ =oil	24	4	18	18

Table 13: Results of the sequential exploration program for the large MRF case study shown in Section 5.2, for single cluster update strategies with different cluster size. $i_{(1)}$, $i_{(2)}$ and $i_{(3)}$ are respectively the first, the second and the third best reservoir unit selected.

For sequences (Table 13), the cluster size has a greater influence than in the BN case of Section 5.1, since the best nodes are now more spread out in different clusters. The first best pick is a typical myopic first best pick and corresponds to reservoir unit 14. If this is oil, we remain in the same cluster, and we go for unit 13. If it is dry, the algorithm suggests to move to unit 10 in the 4-cells cluster configuration and to unit 24 in the 2-cells cluster configuration. This happens because the GI of cluster 4 in the 4-nodes clustering is influenced by the presence of two almost-sure dry nodes at the bottom, while cluster 7 in the 2-cells clustering has a good GI, since it is made just by two nodes whose presence of oil is quite likely. In general, the 4-nodes clustering strategy shows a better ability to test new areas and to come back to the more certain places in case of dry discoveries. If unit 10 is oil, we remain close and drill unit 20, while if it is dry we move back to unit 24. The 2-cells clustering shows, on the other side, less adaptation, for example in the suggestion of drilling unit 24 no matter the outcome of unit 13, due to the small size of its clusters and to the absence of updating given by the SCU strategy. RHLA Dpt 1 and Dpt 2 strategies do not have the constraint of the clusters and therefore their behaviour is more flexible: the first two nodes belong to different zones of the field, no matter the outcome of the first choice. At the third step, if both unit 14 and 19 are found dry we move to unit 40, exploring a third new area of the reservoir.

In practice the petroleum companies tend to target the most lucrative units at any time. This example shows that it may be better in the long run to balance the search over the dependent prospects.

6 Conclusions

In this paper we have used clustering of large graphs to suggest dynamic designs of nodes. The main output of the suggested algorithms is approximate sequences of nodes, where the selection of nodes is guided by entropy or profit-based utility functions. In addition, we can compute the expected utility of strategies, with an upper bound. In all the examples that we have tested, the proposed strategies and the proposed bounds perform much better than classical naive or myopic strategies which are commonly used in the petroleum industry today.

Most big oil fields have been discovered. There is a tendency in the oil and gas industry of looking for smaller volumes. In such situations, accurate statistical descriptions of the dependencies are important. Small prospects may not be lucrative marginally. But seen jointly, a set of prospects may be lucrative. Our methods shows that this dependence allows flexibility in the search for drilling targets, and that this view can give clear gain in profits, on average. Since there is much structure in the models for petroleum prospects, the clustering approaches seem promising to deal with the intracable search problem.

The presented methods rely on fast updating of the probabilities of the graphical model. We have outlined one strategy which uses multiple cluster updating at all stages of sequential data collection. This is more time-consuming than a single cluster updating scheme, which does only local updating of probabilities and the cluster-wise Gittins Indeces. In the examples we examined, the multiple cluster updating approach with small clusters is often preferable over the single cluster updating approach with larger clusters. In practice, the comparison depends on the cluster size and the dependence in the probability model. If the clusters are small and the learning within the cluster is marginal, or comparable with the learning provided by other clusters, the multiple cluster updating is favourable. In cases where the learning within the cluster is higher, the single cluster updating performs well (See Section 4.2, right network).

The clustering approach is most useful when there is much structure in the graphical model (parts of the network are almost uncorrelated with other branches). With lack of structure, two challenges arise: i) the selection of clusters gets harder and more sensitive, ii) the learning between clusters can be high and the efficiency of the clustering approach goes down. For instance: The left network in Section 4.2 with little structure shows that it is difficult to assign correctly the nodes to each cluster. Our partition here might be suboptimal. Still, there are too many possible orderings of clusters to search for the optimal one. Ideally, the guideline should be to create clusters composed by nodes that share some common feature, like a geographical proximity (MRF examples) or some known correlation based on previously identified covariates (BN example of Section 5.1). In cases where there are no such intuitive directions for clustering, we suggest to use trial and error over a couple of clustering sets.

We compared the clustering approach with the look-ahead algorithms, which do not explicitly use the structure of the graph. There were clear improvements when using clustering on the structured graphs (like the Markov chain), but no added value of clustering in unstructured graphs like the common parent network in Section 4.2.

The Monte Carlo sampling approach for approximating the expected utility can have large variability. One may improve some of this with variance reduction techniques. The upper and lower bounds can be computed without Monte Carlo when the discounting factor $\delta = 1$. In most applications the sequentiality is influenced by δ, and improved sampling-based strategies would be useful.

Many design problems are different in the sense that a fixed number of nodes can be selected at once, without being able to modify the node selection after observing the first node(s). For instance, a petroleum company may decide to drill two wells at once, without learning, simply

because the equipment is available and money is saved. This entails another discrete optimisation problem, but some of the presented cluster ideas might be re-used here.

We have not considered budgetary constraints in the current paper. The design can be over as many nodes as is profitable in terms of the utility. It would be interesting to study constraints in the sense that only $N^\star < N$ nodes can be selected. Another related topic we did not study is the situation with imperfect information, i.e. when the decision nodes are only observed indirectly. This amounts to another layer in the graphical models considered here. The design sequences may show more flexibility when both imperfect and perfect data collection is possible. There are several interesting problems at the interface of statistical modelling and inference and operations research / decision making. New insights in such problems will be useful for policy making.

7 Acknowledgments

We thank the Statistics for Innovation (SFI^2) research center in Oslo, that partially financed GMs scholarship through the *FindOil* project. We acknowledge Arnoldo Frigessi and Ragnar Hauge (Norwegian Computing Center) and David Brown and Jim Smith (Duke University) for very interesting discussions on this topic.

References

Aalders, I., Hough, R. L. and Towers, W. (2011). Risk of erosion in peat soils an investigation using Bayesian belief networks. Soil Use and Management *27*, 538–549.

Abdul-Razaq, T. S. and Potts, C. N. (1988). Dynamic Programming State-Space Relaxation for Single-Machine Scheduling. The Journal of the Operational Research Society *29*, 141–152.

Benkerhouf, L., Glazebrook, K. and Owen, R. (1992). Gittins indexes and oil exploration. Journal of the Royal Statistical Society. Series B *54*, 229–241.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society, Series B *36*, 192–236.

Bhattacharjya, D., Eidsvik, J. and Mukerji, T. (2010). The Value of Information in Spatial Decision Making. Mathematical Geosciences *42*, 141–163.

Bickel, J. and Smith, J. (2006). Optimal Sequential Exploration: A Binary Learning Model. Decision Analysis *3*, 16–32.

Brown, D. and Smith, J. (2012). Optimal Sequential Exploration: Bandits, Clairvoyants, and Wildcats. submitted, accessible at http://faculty.fuqua.duke.edu/ jes9/bio/OptimalSequentialExplorationBCW.pdf .

Chen, Y. R. and Katehakis, M. N. (1986). Linear Programming for Finite State Multi-Armed Bandit Problems. Mathematics of Operations Research *11*.

Claxton, K. and Thompson, K. (2001). A dynamic programming approach to the efficient design of clinical trials. Journal of Health Economics *20*, 797 – 822.

Cowell, R., Dawid, P., Lauritzen, S. and Spiegelhalter, D. (2007). Probabilistic Networks and Expert Systems. Springer series in Information Science and Statistics.

Gittins, J. (1979). Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society, Series B *41*, 148–177.

Glazebrook, K. and Boys, R. (1995). A class of Bayesian models for optimal exploration. Journal of the Royal Statistical Society. Series B *57*, 705–720.

Krause, A. and Guestrin, C. (2009). Optimal Value of Information in Graphical Models. Journal of Artificial Intelligence Reseach *35*, 557–591.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. Journal of the Royal Statistical Society, Series B *50*, 157–224.

Le, N. and Zidek, J. (2006). Statistical Analysis of Environmental Space-Time Processes. Springer.

Marcot, B., Holthausen, R., Raphael, M., Rowland, M. and Wisdom, M. (2001). Using Bayesian belief networks to evaluate fish and wildlife population viability under land management alternatives from an environmental impact statement. Forest Ecology and Management *153*, 29–42.

Martinelli, G., Eidsvik, J. and Hauge, R. (2011a). Dynamic Decision Making for Graphical Models Applied to Oil Exploration. Mathematics Department, NTNU, Technical Report in Statistics *12*.

Martinelli, G., Eidsvik, J., Hauge, R. and Drange-Forland, M. (2011b). Bayesian Networks for Prospect Analysis in the North Sea. AAPG Bulletin *95*, 1423–1442.

Martinelli, G., Eidsvik, J., Hauge, R. and Hokstad, K. (2012). Strategies for petroleum exploration based on Bayesian Networks: a case study. SPE Paper 159722, accepted for presentation at the 2012 SPE ATCE conference, 8-10 October 2012, San Antonio, TX, USA .

Powell, W. (2007). Approximate dynamic prgramming: solving the curses of dimensionality. Wiley.

Puterman, M. (2005). Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley's Series in Probability and Statistics.

Reeves, R. and Pettitt, A. (2004). Efficient recursions for general factorisable models. Biometrika *91*, 751–757.

Shewry, M. C. and Wynn, H. P. (1987). Maximum entropy sampling. Journal of Applied Statistics *14*, 165–170.

Tjelmeland, H. and Austad, H. (2012). Exact and approximate recursive calculations for binary Markov random fields defined on graphs. Journal of Computational and Graphical Statistics *doi: 10.1080/10618600.2012.632236*.

Wang, Q. R. and Suen, C. Y. (1984). Analysis and Design of a Decision Tree Based on Entropy Reduction and Its Application to Large Character Set Recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on *PAMI-6*, 406 –417.

Weber, J., Sun, W. and Le, N. (2000). Designing and integrating composite networks for monitoring multivariate Gaussian pollution fields. JRSS Series C *49*, 63–79.

Whittle, P. (1980). Multi-armed bandits and the Gittins index. Journal of the Royal Statistical Society. Series B *42*, 143–149.

Zidek, J. and Zimmerman, D. (2009). Monitoring network designs. in Handbook of Spatial Statistics (ed. A. Gelfand, P. Diggle, M. Fuentes and P. Guttorp) , 131–148.