Dynamic exploration designs for graphical models using clustering

Gabriele Martinelli and Jo Eidsvik

Department of Mathematical sciences, NTNU, Norway

Abstract

The paper considers the problem of optimal sequential design for graphical models. 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. The goal is to maximize utility, here defined via entropy or total profit. With a large number of nodes, the optimal solution to this selection problem is not tractable. Here, an approximation based on a subdivision of the graph is considered. Within the smaller 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 value of the strategy. 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 guided by cluster values can provide useful guidelines to policy makers.

1 Introduction

Our interest here is a sequential selection problem over a 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 we consider the challenge of constructing an optimal dynamic or sequential design of nodes. 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 will be updated, according to Bayes rule, and these updated probabilities depend on where and what we observe. Relevant design questions are then: Which nodes are more informative? Which sequence of nodes gives the best policy?

At each stage, we can choose to observe one additional variable, or quit the search. If we acquire data at a node, we apply Bayes rule to 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. With independent variables, the only focus would be part i), where we go for the most lucrative variables, measured by the utility criterion. Within our context of dependent variables, represented by a graph, knowing key variables gives us the chance to make better, informed, decisions at the later stages of the sequential strategy. The continuation value (CV) part in ii) then plays an important role in the utility function.

The 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.

There is a large literature on constructing approximations to DPs in high dimensional problems. See e.g. Powell (2007). The appropriate solution seems to be very case-specific. Martinelli et al. (2011a) proposed an approximate solution to the oil and gas exploration problem. The approach is based on combining DP with heuristics, in order to produce a sufficiently accurate result within a reasonable time. The main problem with such heuristic approaches is the difficulty in assessing the properties of the solution. It is really hard to know how close the heuristic solution gets to the optimal solution. A possible alternative is to utilize the structure of the model. For graphical models one natural approach is to split the original graph in several clusters. This idea was originally presented in Brown and Smith (2012). They next solved the DP exactly in the clusters, and combined the results to get approximations for the expected utilities on the full-size graph. They also computed bounds indicating the quality of the approximation.

Our main focus 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 (GI) developed by Brown and Smith (2012). These 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 oil and gas situation, 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), management project selection, 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 splitting the problem 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 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.

More generally, the sequential design problem we consider here entails a selection of nodes, one at a time, to maximize the expected value of some utility function. We first introduce the statistical

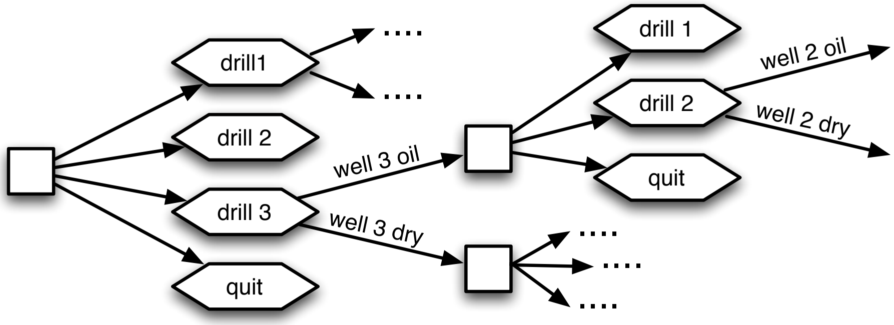

Figure 1: Decision tree for a simple 3-nodes discrete example with two possible outcomes (*oil* or *dry*) per node.

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 $x_i \in \{1,\dots,k_i\}$, $i = 1,\dots,N$ denote the discrete random variables. Without loss of generality, we assume $k_i = k$ possible states for all nodes i. We represent the probabilistic structure via a graph, and $\mathcal{N}$ denotes the set of all nodes. For a BN defined by a directed acyclic graph the joint distribution $p(\mathbf{x}) = p(x_1,\dots,x_N)$ is

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

where 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 have:

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

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

We assume known, fixed, statistical model parameters in $p(\mathbf{x})$, such as β and $\alpha_i(x_i)$ in equation (2) and the conditional probabilities in equation (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 $DV(i) = \max(0, \sum_{j=1}^{k} r_i^j p(x_i = j))$, $i = 1,\dots,N$. 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 updates 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 H. Tjelmeland and Austad (2012).

Assume we can acquire data at one node in the graph, and use this outcome to update the prior model. In this way we incorporate the evidence 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 $i = 1, \dots, N$. 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, $\omega_i = 1$ means 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 $\boldsymbol{\omega} = \boldsymbol{\omega}_0 = (-, \dots, -)$. 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 $\boldsymbol{\omega} = (-, 1, -, \dots, -)$. 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 $p(x_i = j|\boldsymbol{\omega})$, $i = 1, \dots, N$, $j = 1, \dots, k$. 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). As discussed above, 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.

The sequential design procedure forms a decision tree, where a fork represents a decision to choose a node, 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 sequences to construct the decision tree.

2.2 Dynamic Programming for Sequential Design

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 of future cash flows, given that we are in observation state $\boldsymbol{\omega}$. Initially, the vector of observables is empty: $\boldsymbol{\omega}_0 = \{-, -, \dots, -\}$, and the value is $v(\boldsymbol{\omega}_0)$. The goal is to compute $v(\boldsymbol{\omega}_0)$ and to find the associated optimal sequential design. The solution is given by DP, where we as an integral part evaluate the values $v(\cdot)$ for any possible combinations of sequential evidence.

Recall that we always have the option of quitting, when the expected future revenues are negative. We introduce a discounting factor δ. In practice, a δ near 1 encourages learning the dependent model, while a smaller δ means that we choose the bigger IVs at the early stages. 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. For short, we may write this as an immediate gain and a CV to get

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

where $\boldsymbol{\omega}_i^j = \{\boldsymbol{\omega}_{\textbf{-i}}, \omega_i = j\}$ 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 performing sequential selection. Entropy is not directly related to cost, revenues, or phrased in a decision theoretic setting. Nevertheless, it has become a very useful tool to construct designs, especially when there is no obvious criterion for an ultimate decision.

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 again introduce a 'price' P_i of observing node i. This is now fixed, no matter the 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

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

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 reduction in entropy. An example is presented in the next section.

Note that this 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 what method we use for 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 simple motivating example

We 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 ($k = 2$). Related to the oil and gas situation, we refer to these two by 'oil' and 'dry'.

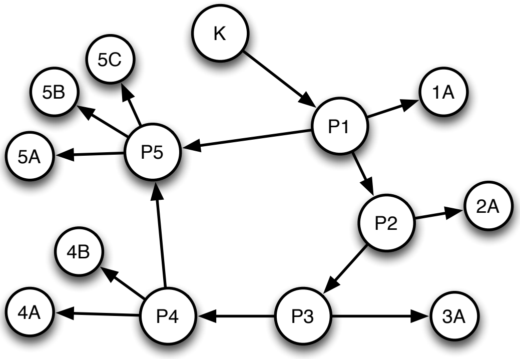

Figure 2: Simple example used in Section 2.3. We can collect data in the leaf nodes. By Dynamic Programming we construct optimal sequential designs that maximize expected utility.

The main input parameters for this example are given in Table 1. Based on cost/revenues only two nodes, 3A and 4A, have positive marginal expectation or intrinsic values. 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 $\delta = 0.99$.

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
Marginal 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
Intrinsic values	-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.

An optimal decision maker, using equation (3) and equation (4), would account for the ultimate consequences of the actions.

Results are shown in Table 2. We here compare the outcomes of the naive and myopic strategies with the optimal using DP.

A myopic strategy is based on forward selection. Using cost/revenue utility, this strategy starts from the most lucrative prospect 3A. If this variable is dry, we update the network and find out that all the DV 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 increases significantly. In fact, six of the seven remaining DVs are positive. The myopic approach goes for the greatest of all DVs, and we select 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.

A optimal sequential strategy based on costs/revenues privileges the nodes that are valuable in terms of prospective learning, but also profitable. While the sub-optimal myopic approach only looks forward, the DP solution is a forward-backward algorithm. It accounts for all possible future data collection. The values $v_i(\boldsymbol{\omega}_0)$ (see Equation 3) are the following: $[3352, 3952, 3595, 3427, 3852, 3926, 3443, 3738]$. We can immediately see two important features: i) all these values are much bigger than the naive value of the field, and this is natural since the correlation in the graph is high. ii) there is not much difference between the highest value (prospect 2A) and the lowest one (prospect 1A), since the discounting factor δ is very close to 1. This small gap, though, is sufficient to define a sequence for optimal design.

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). In the event of 2A oil: If 3A is dry, we again go to the other end of the network (5A). If 3A is oil, we remain in the same area (4A). Note that in spite of its high DV, prospect 4A is not selected until the third stage, and only after confirming oil in the neighboring node 3A. This means that node 4A does not carry so much information about the other prospects.

For the entropy-based design, we again compare a myopic strategy with a full DP based strategy. In a myopic strategy the first node selected would be 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 expect to move away from the 5-nodes, since most of the uncertainty in that part of the graph has been resolved. This is exactly what happens: If node 5B is dry we move to node 2A. If it is oil, we move to node 4A.

In a DP entropy-based strategy, 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 $v_i(\boldsymbol{\omega}_0)$ for $i = 1, \ldots, 8$ are as follows: $[0.8534, 0.8487, 0.8066, 0.7850, 0.7713$,$0.8520, 0.8353, 0.8353]$. Nodes 4A and 4B, which are characterized 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. 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 t *quit* the strategy.

3 Clustering strategies for large graphs

The optimal sequential design relies on DP. For large graphs the number of possible scenarios to evaluate exceeds what is computationally tractable. The DP utility functions, e.g. (3), 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 various utilities and cluster configurations, and use 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, $d = 1, \dots, L$ be disjoint nodes of the original graph $\mathcal{N}$, 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. Just like for DP, the clusters cannot be much larger than ten for the suggested method, since the computing explodes and the number of scenario breaks the memory in the computer. The approximations we present here improve as the cluster sizes get larger, 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 a cluster. The ranking is based on DP within clusters, given the current information. Once we collect data in a cluster, we update the probabilities, 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 indix (GI), see Gittins (1979) and Whittle (1980). We consider the cluster-wise GIs in the spirit of Brown and Smith (2012). First, we introduce a variation of equation (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:

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

Now, when the computation is restricted to cluster C^d, the GI for cluster C^d in state $\boldsymbol{\omega}_{C^d}$ is $M_{C^d}(\omega_{C^d})$, defined as the smallest retirement value M such that $v^d(\boldsymbol{\omega}_{C^d}, M) = M$. 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 expression (eqGI).

Brown and Smith (2012) derived some important properties for the value function $v(\boldsymbol{\omega}, M)$, for any evidence $\boldsymbol{\omega}$. Figure 3 illustrates the value functions for one of the examples below. Here, we plot $v(\boldsymbol{\omega}_{C^d}, M) - M$ for some clusters, and 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. This means that the optimal cluster may change for different retirement values. This is an important feature that links GIs and the value of information, and enforces DPs ability to read 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

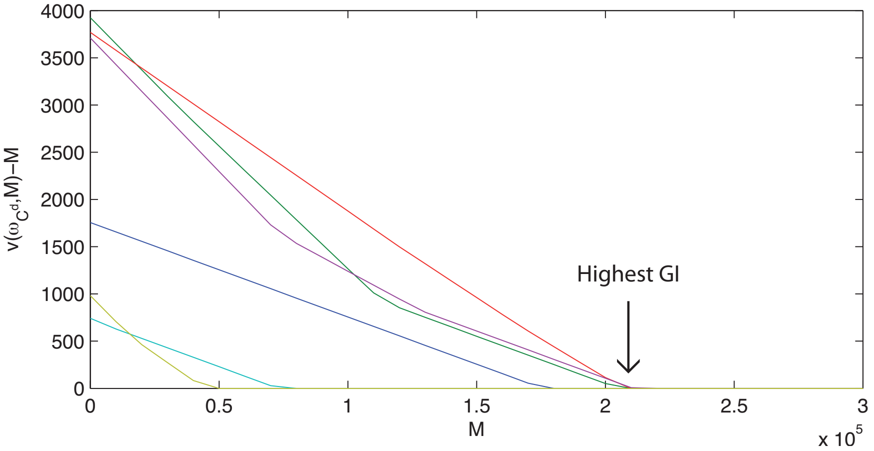

Figure 3: Values of $v(\boldsymbol{\omega}_{C^d}, M) - M$ for 6 clusters of the 13 shown in Figure 9; the highest Gittins indeces are achieved by clusters 4, 5 and 8.

DP for the clusters. Since the value function is piecewise linear, solving equation (3.1) for fixed M is relatively fast. Alternatively, one can use theory from Markov decision processes to tranform the DP into a linear programming problem, see Chen and Katehakis (1986) and Brown and Smith (2012).

Instead of going through all branches of the decision tree resulting from this strategy, we will sample outcomes of all variables in the graph. This kind of Monte Carlo sampling can also be used to evaluate the sequential strategy (Section) . This entails running a number of hypothetical scenarios where we sequentially observe (i.e. sample the outcomes) of the chosen nodes, update the probabilities, and proceed to the next stage.

Note that the updating step can be quite time-consuming. In principle, the evidence vector $\boldsymbol{\omega}$ in equation () is the full observation sequence until this stage. This means that all cluster probability models must be updated when we acquire new data at a node. A faster, but more approximate strategy is to update only the cluster where the data is collected. This means that only one of the GIs changes at each stage. We implement both of these different methods for integrating the sequential observations.

- **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 in the selected node. All cluster GIs are also updated based on the updated probabilities. By updating we mean recomputing the joint probability distribution and the corresponding GI. Then, we choose the best cluster at the second stage using updated values for GIs and updated DPs. 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 efficient since we update the entire graph whenever a new piece of 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. On the other side, the SCU method is faster since the only cluster that needs to be updated is the one where current information has been gathered. However, the sequential design from SCU could suffer lack of accuracy. Pseudo-algorithms constructing sequences over Monte Carlo samples are described in Algorithm 1 and 2. Both methods can be used not only to derive sequences, but also to approximate the DP value. For this task, we have to sum the values $t_{s_{C^*}}$ selected at every step of the `while` cycle, possibly discounting them with a factor δ (in this case the computation of $v(\boldsymbol{\omega})$ must also keep into account δ!).

Algorithm 1 Evaluating a Single cluster Update strategy

```
$\boldsymbol{\omega} = [-, -, \dots, -]$                          # 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 $C^*$
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 value $v(\boldsymbol{\omega}_0)$. Of course, the clustering strategy gives a sub-optimal value compared to the full DP solution, which is not tractable for high dimensions. A most 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 the value $v(\boldsymbol{\omega}_0)$. The Monte Carlo strategies in algorithms 1 and 2 provide a sampling-based approach for estimating the intial value. Here, each Monte Carlo sample constructs a design sequence which depends on the outcome at the selected nodes. By averaging over B samples from the graphical model, this gives a Monte Carlo estimate for initial value. The values will differ between MCU and SCU, since the full updating scheme gives

Algorithm 2 Evaluating a Multiple clusters update 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 $C^d$
    $GI_{C^d} = M : \; v(\boldsymbol{\omega}_{C^d}) - M = 0$          # Updated GI for cluster $C^d$
  end for
end while
```

less 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 on the evidence vector $\boldsymbol{\omega}$ restricted to cluster d:

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

Then, a lower (independent) bound for the expected utility is the sum of the marginal values for all the clusters:

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

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

The independent value can be improved by using the outputs from sequential cluster selection. Assume we start by evaluating the cluster with the highest GI. Its value is $v(\boldsymbol{\omega}^{C^d})$. We next generate an outcome for this cluster $\mathbf{t}_d$, and we run DP restricted to this cluster, plugging in the sampled data at the selected nodes. We next update the probability model at the remaining $L-1$ clusters, and choose the cluster with highest GI, say C^e. We use DP and sampling for this cluster, update, and so on, until all the clusters are considered. The average value of this lower bound, over B samples, is

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

where the conditioning is the empty set for $d = 1$, and $\mathbf{t}_b^{C^{e<d}}$ is the b^{th} sample restricted to the clusters previously considered. Note that the sequential values $v(\boldsymbol{\omega}^{C^e} | \mathbf{t}_d)$ impose learning, and this

gives an improvement over the independent bound:

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

The quality of this sequential strategy depends on the choice of clusters and on the 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. Of course, this policy is not feasible, since we are using information that is not available in practice. But, the clairvoyant information means 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 2 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. Assume we have a sample $\mathbf{t}$ from the graph. We compute the value of a cluster, given all information outside the cluster. We are in this way pretending to know everything happening around us. This calculation requires the full conditional for the cluster, 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 the all the clusters different from C^d We can improve our inequality chain to get

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

4 Synthetic examples

We first study small BNs and MRFs where we 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 wanted, the current approach should be interesting on such real cases.

To obtain a better understanding of the DP defined in equation 4, we run it on two small BN, shown in Figure 4. The two BNs are small clusters of a bigger network, connected through a Common Parent (CP) node. Both BNs have 5 nodes that represent sites or variables that can be selected. In the first network the structure is made by a common node and 4 children, while in the second network the structure is linear with two chains departing from a common top node.

Let us start with the network on the left. Each node is binary, with two states that we will indicate as A and B. The top node has a symmetrical prior probability distribution, with 50% chance of being in state A and likewise of being in state B. Nodes 2 and 3 have a CPT with propagation

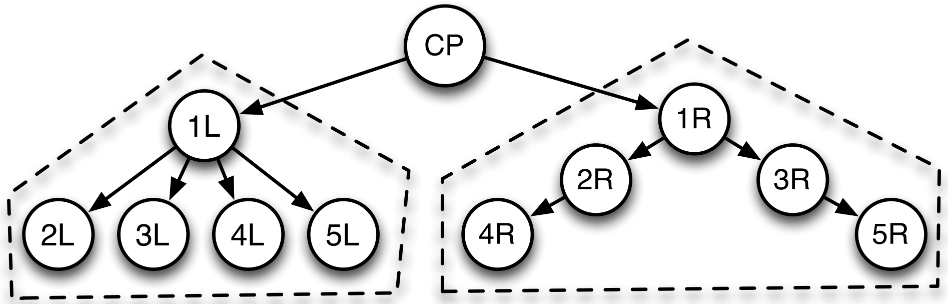

Figure 4: Simple BNs used for testing the entropy criterion, connected through a

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

$x_4, x_5 \ \backslash \ x_1$	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

of information just through state A, while nodes 4 and 5 have perfectly balanced CPT, as shown in Table 3

The original entropy of the network in configuration $\{-,-,-,-,-\}$ is 2.3615 . We intuitively may expect that the reduction in entropy is higher for node 1; actually, as we can see from Table 3, since the probability distribution in node 4/5 is symmetrical, we observe the same reduction for both node number 1 and nodes 4/5. In particular, the entropy is substantially reduced if we observe the state A in either of these nodes (in such case a single configuration, $\{A, A, A, A, A\}$ collects the 65% of 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 of them with a little more than 20% conditional probability of occurrence. The situation is different is we observe A in nodes 2/3: here, the configuration $\{A, A, A, A, A\}$ is still the most likely, but with a mere 46% of occurrence since we are more uncertain about the occurrence in prospect 1 than with a corresponding observation in nodes 4/5. Finally, observing B in node 2 (or 3), there are two equally most likely configurations, namely $\{B, B, A, B, B\}$ and $\{B, B, B, B, B\}$, since the marginal likelihood of a B observation in node 1 in increased, but this has no effect on the conditional distribution for node 3.

The overall effect is that an observation in 1 or 4/5 produces an average decrease in entropy of 0.6931, higher than the reduction brought by an observation in nodes 2 or 3, equal to 0.6109. This is consistent with the intuition.

The question now is, are the results consistent with the intuition even when a full DP strategy is taken into account, i.e. when we have the possibility of keep on drilling until the reduction is higher than a certain cost C? Results (final values for all the nodes) are in Table 4.

We can immediately see some surprising results: in the myopic case, the first best choice is determined just 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 which is the cost of the entropy. If the cost is higher than 0.6931 no node is profitable, because that is the maximum reduction in entropy that we can possibly achieve with a single observation.

The situation when taking into account DP strategies is more complex: if the cost is very high

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

Table 4: Final values of the DP and Myopic strategies applied to the network on the left in Figure 4, for different values of C, and for $\delta = 1$.

(0.65), than we might have to stop after a single observation, at least if the reduction after the first observation is already consistent. That's why node 1 is selected as best choice. We have to remember here, that though the average reduction for node 1 and nodes 4/5 is equal, the marginal entropy of $\{A, -, -, -, -\}$ is different (namely smaller) than $\{-, -, -, A-\}$, therefore where an observation A is 1 could be sufficient, an observation A in node 4/5 could not be sufficient: this reflects in the different final values. When the cost is medium (0.5), node 4/5 are selected first. Here we choose to observe usually 2 or 3 nodes (depending on their outcome), and in such case starting in 4 or 5 is optimal. When the cost is very small (0.2) it is almost always convenient to keep observing up to end, and this makes the values for nodes 2/3 and 4/5 optimal and identical (since the discounting is 0). The only exception is when starting in node 1. In such case we might stop one step before, and the effect is an overall smaller reduction in entropy. Is this necessarily bad? It is true that we have achieved a smaller reduction, but with a smaller number of observation! It is likewise true, though, that if the criterion is "keep collecting evidence until the marginal reduction is smaller than the cost C", the strategy of starting in 2/3 or 4/5 is optimal, as identified by the algorithm.

One may be interested in asking what would happen if $\delta < 1$. The first answer is that it is sufficient $\delta = 0.9$ for solving the ambiguity when $C = 0.5$ in favor of nodes 4/5. The second answer is that in order for node 1 to emerge as a clear winner even when the costs are medium (C=0.5), we need to penalize strongly the observations after the first one: if we move δ to 0.3 we obtain a higher final value for node 1 vs nodes 2/3 and 4/5.

In order to understand if the role of node 1 is really central, we have ran the same strategies on the network on the right in Figure 4, and we have imposed for all the nodes symmetrical CPT, such that the marginal reduction in entropy is now equal to 0.6931 for all the 5 nodes. Results are in Tables 5.

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

Table 5: Final values of the DP and Myopic strategies applied to the network on the right in Figure 4, for different values of C, and for $\delta = 1$.

For this choice of the discounting factor the results for the three nodes are equal in the myopic strategy (as expected, since the marginal reduction in entropy is the same) and in DP with high costs (0.65), since in this case just an observation is allowed; with higher discounting even when costs are medium (0.5), the three results are identical and just one observation is allowed. In every other case it is optimal to start far from the center, since when 2 or more observations are allowed the reduction in entropy achieved by observing, say, nodes 4 and 5 is higher than the reduction achieved by observing 1 and any other node.

When considering the full network, we can consider the two subnetworks as two different clusters. The first cluster (left subnetwork) results in an average case ($C = 0.5$, $\delta = 0.9$) the one with the highest GI, therefore it is selected first. This should not be surprising, since we have already seen that this is a more complex case with a total entropy of 2.36, higher than the tool entropy of the second cluster, equal to 1.99. Within this cluster, the nodes 4 or 5 are the most valuable, as confirmed also in Table 4 (the values are slightly different since $\delta = 0.9$, but the ranking does not change). After observing either of these two nodes, cluster 2 becomes more valuable, and the decision maker therefore should move there to find a prospect that reduces the entropy. The suggested nodes are again 4 or 5, as indicated also in Table 5, on the left. The choice is in this case independent 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 just slightly reduced, just enough to make cluster 2 more valuable in terms of GI. The strategy suggests to keep alternating nodes from cluster 1 and from cluster 2, following the same criteria.

This small example is synthetic, but similar situations may arise in practical real problems in fields such as biology, fishing, and natural resources collection in general. We can think that the two sub clusters are different areas that share little information, and we have to place monitor stations in order to maximize the entropy reduction of our data. In this situation, the proposed approach would suggest where to place the first stations, following a sequential approach.

4.2 Small BN: revenues/ cost

We focus our attention on two small BNs (Figure 5) with 12 correlated nodes; in this setting (revenues/costs based utility function) we will often use the word *prospect* in stead of node, since the most natural application is in petroleum explorations. The small dimension allows an exact solution of the problem. The idea is to show how a different network structure influences the quality of the bounds, and to compare these bounds with the ones produced by approximated sequential strategies presented in Martinelli et al. (2011a).

The first case study (on the left in Figure 5) 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 also Martinelli et al. (2012).

We are first interested in studying the outcome of sequential strategies under different clustering configurations. It is important to keep into account Table 8, where the marginal probabilities and the intrinsic values for all the 12 prospects are stored. Let us assume we are still in an oil exploration setting, with two possible outcomes, *oil* and *dry*, and for simplicity let us consider here strategies SCU. We have few prospects with positive IV, the most prominent being number 8, 10 and 11. In spite of this, the exact DP procedure (see Table 6) select node 5 as the best candidate for the 1st choice.

The interpretation is not immediate, since we are running a DP with an horizon of 12 steps, and the consequences of our action can not be visible in the short term. Furthermore, our network is designed in order to allow for both positive and negative correlations between the prospects. This

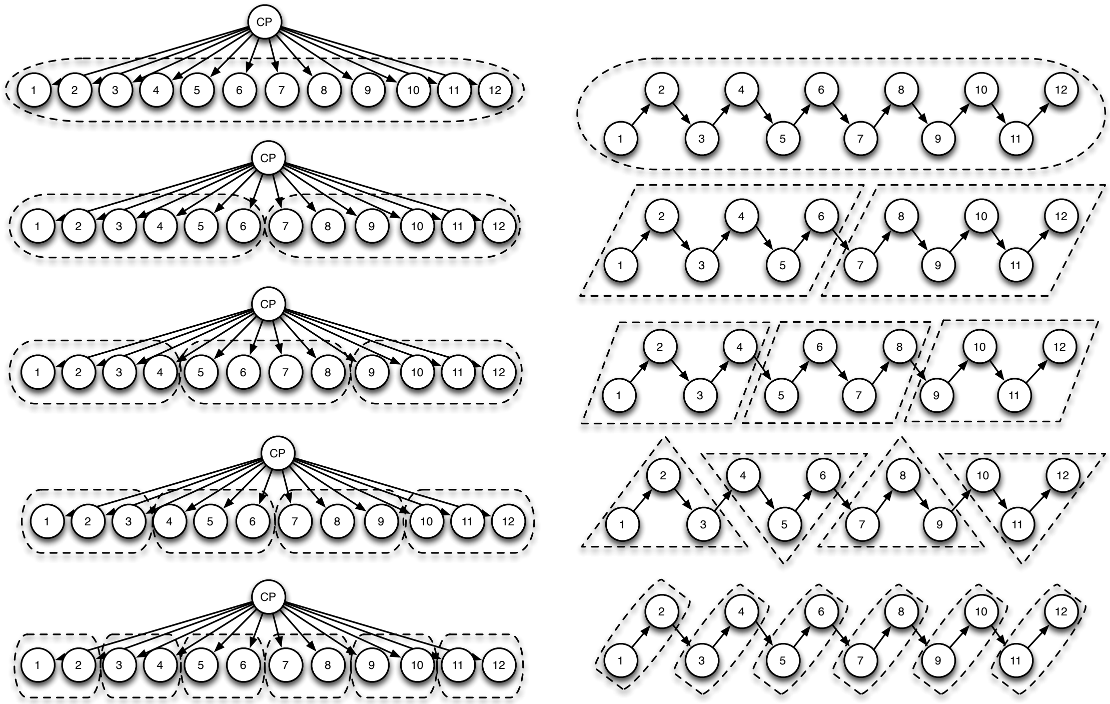

Figure 5: Small BN case studies used in Section 4.2.

# Clusters Cluster size	1 12	2 6	3 4	4 3	6 2	12 1
$i_{(1)}$	8	8	8	8	8	8
$i_{(2)} \vert x_{i_{(1)}} = dry$	10	10	10	10	11	11
$i_{(2)} \vert x_{i_{(1)}} = oil$	2	11	10	10	11	11
$i_{(3)} \vert x_{i_{(1)}} = dry, x_{i_{(2)}} = dry$	5	11	11	11	7	10
$i_{(3)} \vert x_{i_{(1)}} = dry, x_{i_{(2)}} = oil$	1	7	7	7	7	10
$i_{(3)} \vert x_{i_{(1)}} = oil, x_{i_{(2)}} = dry$	5	10	11	11	10	10
$i_{(3)} \vert x_{i_{(1)}} = oil, x_{i_{(2)}} = oil$	11	5	11	11	10	10
Average Value	7295	6992	6779	6638	6398	5783

Table 6: Results of the sequential exploration program 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 node selected. Q means *quit* (the strategy).

means that, for example, finding 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, as shown in the CPTs in Table 7.

This is why when we consider the second column of Table 6 (2-clusters approach), if prospect 8 is found initially dry, we choose prospect 10, that is negatively correlated, while if prospect 8 is found oil we move to prospect 11, that is positively correlated, and so on for the third best choice. The difference in the first choice between the 1-cluster and the 2-clusters strategies is explained by observing that the second cluster (prospects 7-12) has a higher IV and also a higher GI with

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

respect to the first cluster, and therefore it is immediately selected. In general we observe a much less flexibility in strategies characterized by large clusters, and therefore a resulting smaller final value. It is interesting to notice that the strategy with 12 clusters, played with the approach SCU, coincides in fact with the naive strategy, and the first 3 selected nodes are 8, 11 and 10, no matter their outcome.

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

The most important finding is that, when strategies are actually played on real samples drawn from this network, the final results, shown in the last line of Table 6, support and confirm the theoretical results of Table 9. In this case the results are computed by averaging and discounting the final revenues on 100 samples drawn from the network and played with each of the considered strategies. These results are bounded by the estimated Lower and Upper Bounds computed in Table 9. The value for the 2-clusters approach, that looks higher than the exact DP strategy with a single cluster, is simply explained by MC variability. The 2-clusters approach reveals a very high efficiency, with a computational time that is several order of magnitudes less than the SCU.

# Clusters	1	2	3	4	6	12
Cluster size	12	6	4	3	2	1
Independent value (LB1)	-	6660	6177	6115	6043	5056
Sequential value (LB2)	-	6953	6980	6974	6892	6179
Exact DP value	7168	-	-	-	-	-
Clairvoyant value (UB)	-	8263	8289	8295	8295	8295
DP Time per cluster (sec)	6521	0.962	0.073	0.020	0.006	0.001
RHLA Dpt	6114	6534	6880	7024	7146	7158

Table 9: Lower and upper bounds with clusters of different size, for the case study presented in Section 4.2

As we have remarked in Section 3.2, one of the main problems is that in many cases we don't know how far these approximations are from the real value, especially when it is not possible to compute the exact DP strategy because of the size of the problem. For this reason we have studied the possibility of bounding the value of the strategy, for different clusters' configurations.

Let us begin from the example that we have used before, shown on the left in Figure 5.

As we can see, in this case there is nothing that suggests a good hint for placing the clusters, therefore we just follow a naive order when defining different clusters. The results show that without a natural clustering, it is difficult to assign correctly the nodes to each cluster. In a way, choosing $\{1, 2, 3, 4, 5, 6\}$ and $\{7, 8, 9, 10, 11, 12\}$ as the members of the first partition might be suboptimal. Further, it might be even less optimal to build the sequential bound in a random order. In this small case, though, there are $\begin{pmatrix} 12 \\ 6 \end{pmatrix}/2 = 462$ possible ordering of the first two clusters, and we have potentially to test all of the them before deciding which is the optimal partition. We report all the results in Table 9 and in Figure 6 on the left.

The results show that the sequential value (LB2) remains quite close to the optimal value, even for relatively small clusters. The independent value (LB1) increases with increasing clusters' 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, and it is proved by the fact that the gap between the sequential and the upper bound is 1421 for clusters of size 3, and 1321 for clusters of size 2. This is related to what previously said about the absence of a natural clustering criterion.

We have also tried to compare these results with similar results obtained via Rolling-Horizon Look-Ahead (RHLA) strategies, presented in Martinelli et al. (2011a). These strategies approximate the DP value after a certain number n of steps with heuristics approximations, resulting in what we have defined as *Depth n* (Dpt n) RHLA strategies. The values are shown in the bottom rows in Table 9. The columns indicate the depth of the look-ahead procedure in this method. 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 in this case 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).

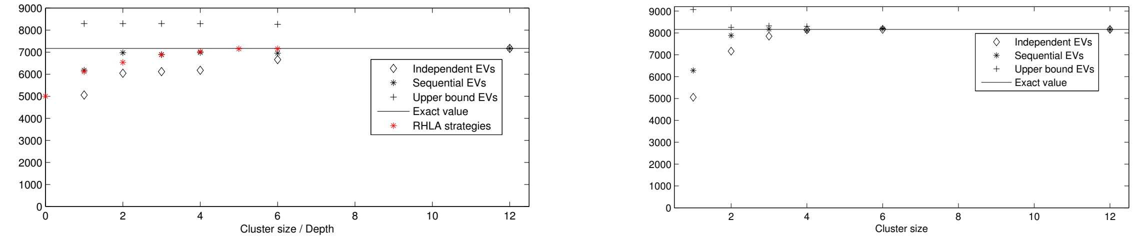

Figure 6: Independent LB, sequential LB, clairvoyant UB and exact values for the two examples shown in Section 4.2; for the first example we show also the final values provided by RHLA strategies.

We now consider the second case study shown in the right in Figure 5. The transition matrices (CPTs) between the nodes are the same used for the previous example, but in this case the covariance structure is imposed through a Markov chain. Thus, there is a predetermined order that makes more natural the choice of the clusters. As we might expect, the bounds are in this case much tighter: in the previous case (common parent structure) we had conditional independence between children only through the common parent, which was not possible to observe, and this made the learning very hard. With the Markov chain structure, on the other hand, we have conditional independence between clusters, given a separating cluster: this suggests a clear hint about the location of the clusters, and it furthermore simplifies the learning process. The results are shown in Figure 6, on the right. In this case the bounds shrink much faster than in the common parent network, and a cluster size of 3 prospects is sufficient to capture virtually all the learning throughout the network. Here, the gap between the sequential and the upper bound is 149 for clusters of size 3, and 371 for clusters of size 2, and this shows that the clustering strategies are much more effective in this kind of scenario.

4.3 MRF, small cases

We have adapted the same ideas originally developed for BN for a Markov Random Field structure. Again, we can imagine that we have a lattice where each node represents a possible prospect, and we are interested in finding the best drilling sequence and approximating the expected value of the whole field. Since solving DP on a medium/large lattice (more than 10 nodes) is not feasible exactly, we can split the lattice in a number of sub lattices, and we can solve the problems in each of these small sublattices (clusters).

We first test our method on a small 3×4 lattice with 12 nodes corresponding to 12 potential prospects. The MRF is an Ising field with 2 colors, representing oil and dry states. Revenues and costs are equal for every prospect and symmetrical ($+3$ and -3 units). The field is non-symmetrical, i.e. marginal probabilities a priori for oil and dry states follow a parabolic trend with a maximum in cell 6 (2nd row, 2nd column). As a direct consequence, the Intrinsic Revenues are oscillating around 0, with positive values just on the left part of the lattice. Marginal prior probabilities and Intrinsic values are shown in Figure 7. The nodes are numbered from left to right and from top to bottom, see any of the examples shown in Figure 8.

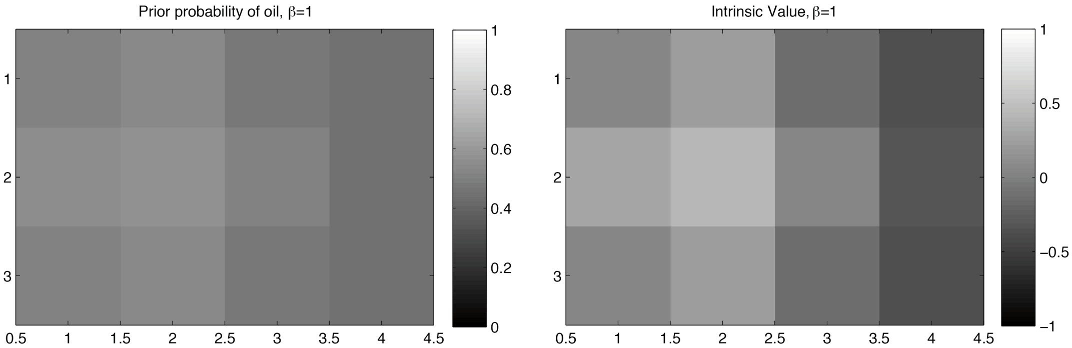

Figure 7: Marginal prior probabilities of oil and Intrinsic Values for the case study presented in Section 4.3

As done in the previous study, we are here interested in the effect that different clusters' size, shape and different values of the parameter β have on the best exploration strategies and bounds. For this reason we propose in Figure 8 six possible cluster combinations for the lattice under consideration. We first analyze the best sequences computed with SCU for a fixed value $\beta = 1$ and 100 simulations: results are shown in Table 10.

We notice immediately that, given the high correlation present in the field, if the first node is found dry there are few hopes to find other good spots: this is the reason behind the suggestion of quitting (Q) the strategy for the 1-cluster scenario, after the first node (prospect 6) is found

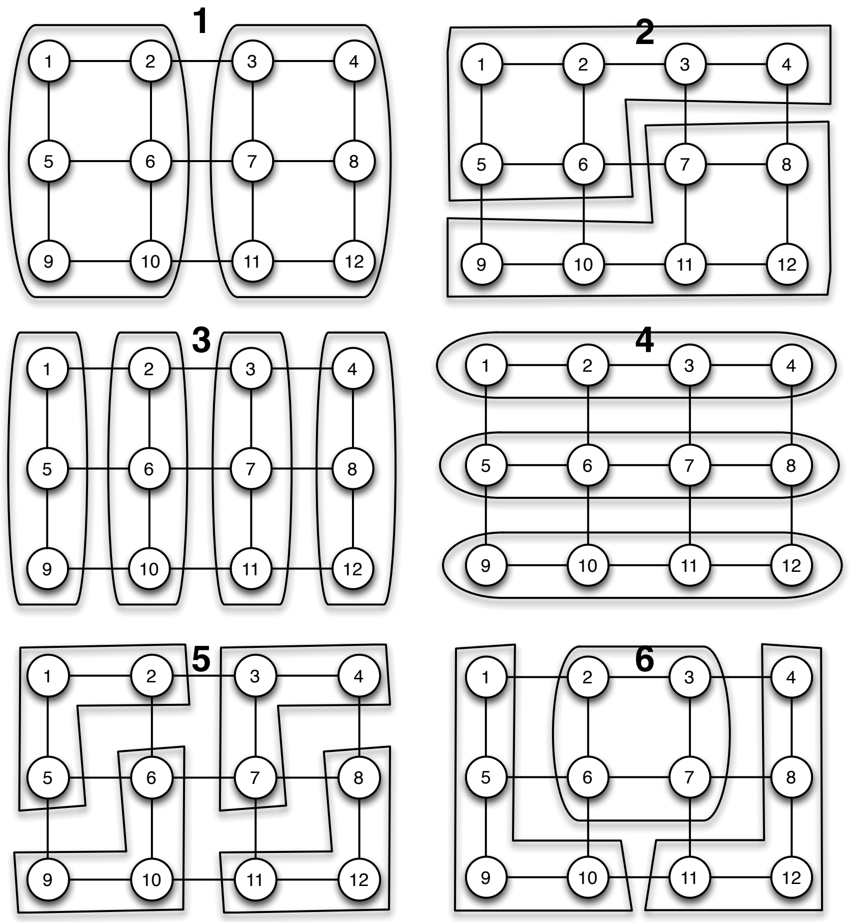

Figure 8: Six possible clusters' configurations for the MRF presented in Section 4.3

dry. Whenever more than one cluster is present, 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, and again the best choice is provided by the 1-cluster configuration, since the suggestion of drilling prospect 7 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 best prospect, namely prospect 6; this reflects the rigid vertical and horizontal clusters present in configurations 3 and 4. As usually, the best way to compare the strategies is to study their effects on a number of fields sampled from the model. The results are in the last row of Table 10 and show average effective values for 100 simulations; for the sake of comparison, we have to remember that the naive value of the field (sum of positive IVs) is 1.305. There is an evident reduction of value when the clusters' number increase, as expected. There is also a less immediate, yet interesting and comforting, increase in value for more compact clusters: 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 is most likely due to 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 10: 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

5.1 Large BN, sequential strategies

The original motivation for this work comes from a large BN describing a geological feature 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 have been 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 idea of dividing the network in sub clusters and studying the corresponding bounds has been presented in Brown and Smith (2012): here, the authors propose, among the others, two possible ways to divide in clusters the original network.

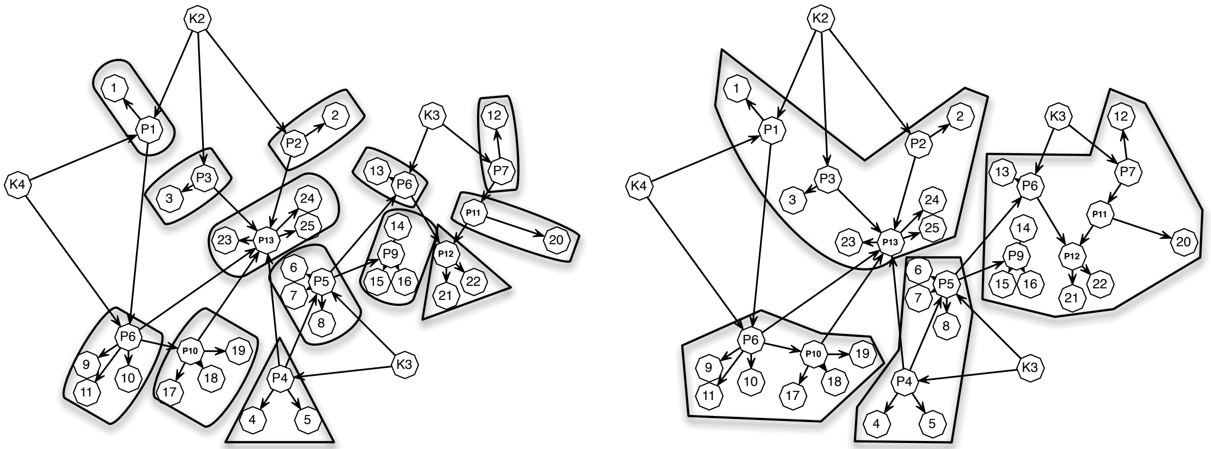

Figure 9: Bayesian Network describing the migration paths of the HC expelled from the source rock. The letter K marks the *kitchens*, i.e. places where the formation of HC has started, the letter P marks the *prospects*, large areas of possible accumulations, while the numbers mark the segments, corresponding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

The first one, with clusters of small dimension, is shown in Figure 9 on the left, while the second one, with clusters of bigger dimension, is presented in Figure 9 on the right. Brown and Smith (2012) describe the effect of different clusters' size on the bounds, and show that the gap is sufficiently small even for small clusters. This is not surprising, given that the learning is limited to few parts of the network and the overall learning is relatively poor; similar comments can be found also in Martinelli et al. (2011a). Now we are interested in studying how the strategies described in described in Section 3.1, for both SCU and MCU, perform on this large network. The results are reported in Table 11.

	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 11: 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 first comment is that the expected revenues increase with increasing cluster size, as expected, and increase when we update every cluster with the new information and not just the cluster where we have collected the piece of information. The difference between the two methods SCU and MCU is stronger when the clusters are small and many, while is less relevant when the clusters and big and few. This should not surprise since in the first case with SCU we may disregard an impact on a large part of the network, while in the second case, since the cluster size is considerable, we are loosing just a peripheral information when using SCU in stead of MCU.

The second consideration is that the average number of nodes drilled increases when going from small to big clusters, but the number of nodes drilled and found dry (that is a good measure of our *accuracy*) decreases, meaning that we are more and 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. In this whole analysis we can not disregard the fact that applying MCU with big clusters is computationally much more expensive than any other option, and that this is often a constraint that the decision maker has to take into account when planning an optimal decision.

When analyzing the sequences (Table 12) we find out that the difference between the SCU and MCU approaches for the small-clusters partition does not appear in the first (obvious) 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 prospect 18, the best prospect, remains the one with highest GI even after removing prospect 18, if the update is positive (oil or gas). Therefore it is selected as second best choice by both 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 prospect oil or gas), we move to a neighboring cluster and we drill prospect 9. In the SCU approach, where the cluster containing prospect 9 has not received the positive information, we pick the second cluster with the a priori highest GI, moving far away towards prospect 8. In the big-clusters partition the same hold, but being now prospect 9 and 18 in the same cluster, we can no longer notice differences between the two approaches, at least in the first 3 choices.

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

Table 12: 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 node selected.

5.2 MRF, big case

In this application we use sequential design on a MRF. The 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. 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 equation (2). The MRF model has 3-colors, where the three distinctions of interest represent respectively *oil saturated sand* ($x_i = 1$), *brine saturated sand* ($x_i = 2$) and *shale* ($x_i = 3$). 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 of 2 Million USD for drilling a dry well (state 2 or 3), while we have a potential revenue of 5 million USD when finding an oil saturated sand (state 1). Before drilling we have the situation represented in Figure 10; here we can see in the top left the marginal probability for the state *oil* in a part of the initial field.

Even if we consider a small subset of the initial dataset (a 4 × 10 square, with 40 potential prospects), the combinatorial complexity 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 use an approach based on clustering in order to show how it is possible to design an optimal strategy and to bound the value of the field. We compare three possible clustering strategies, the first based on 20 very small 2-cells clusters the second based on 10 small clusters of size 2 × 2, and the other based on larger 2 × 4 clusters. The main problem is that since the field is not homogeneous, when we compute the joint cluster probabilities $p(\mathbf{x}_{\mathbf{C(i)}})$ we have still to condition on the entire field, and therefore the computational time required for computing $p(x_{C(i)})$ is proportional to the size of the cluster's sample space. We show the clusters in Figure 10.

Because of the complexity and the size of the field, it is possible to compute with accuracy just the independent lower bound for all the different configurations. Better and more complete results could be obtained by approximating the forward-backward algorithm used for computing $p(\mathbf{x})$, using the arguments presented in Tjelmeland and Austad (2012) We are also interested here of comparing these results with results obtained with different approximations, namely the Rolling Horizon Look-Ahead strategies of different depth presented in Martinelli et al. (2011a) and already used for comparison in Section 4.2. Results are shown in Table 13.

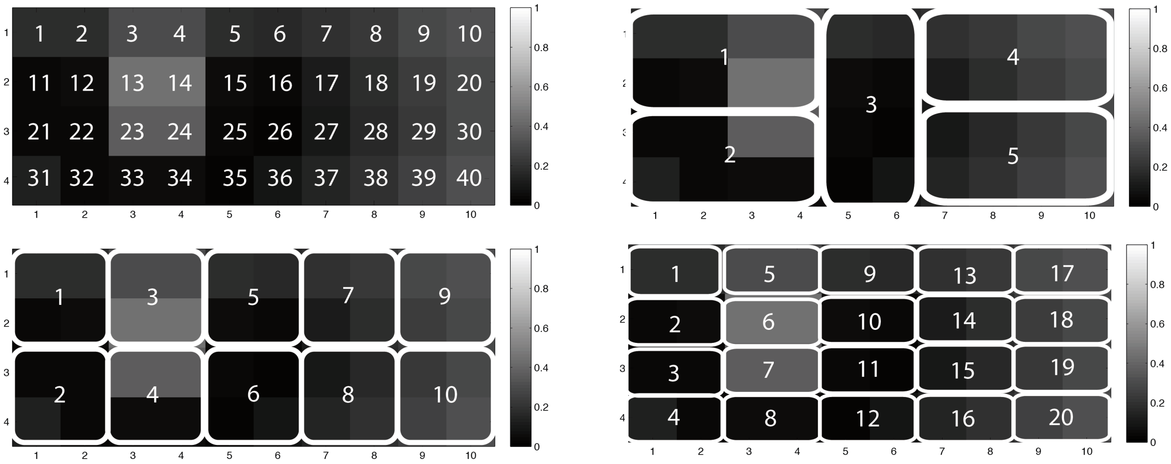

Figure 10: Initial conditions of the MRF described in Section 5.2. Top left: marginal probability for state *oil* in the 10×4 possible prospects. Top right: amplitude seismic data. Bottom left: prior geological knowledge. Bottom right: Probability of oil saturated sand with interaction parameter $\beta = 0.8$.

Cluster size	2-cells	4-cells	8-cells	Naive	Dpt 1	Dpt 2
Independent LB	8.04	10.44	12.17	-	-	-
Sequential LB	9.12	13.71	-	-	-	-
RHLA LB	-	-	-	4.21	8.36	10.74
Clairvoyant UB	28.23	17.00	-	-	-	-

Table 13: 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: $\beta = 1$, $\delta = 0.99$.

We immediately notice how in this case a clustering strategy with large clusters produce better results than the the RHLA strategies until Dpt 2 (Dpt 3 is too expensive to compute). Even simple 2-cells clusters give a much better result than the classical naive approach (sum of positive intrinsic values), and the result is further improved when using 4-cells and 8-cells clusters. In this case the computation of the sequential and clairvoyant bounds has been possible just the smaller clusters and for a number of samples much smaller than the previously considered one. It is worth to notice that the gap between sequential LB and Clairvoyant LB is already quite narrow with 4-cells clusters, and that the improvement is consistent.

When we move to sequences (Table 14), we notice that the cluster size and shape have 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 prospect 14. If this is oil, we remain in the same cluster (we are following a SCU strategy), and we go for prospect 13. If it is dry, the algorithm suggests to move to prospect 10 in the 4-cells cluster configuration and to prospect 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. Iif

	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 14: 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 node selected.

prospect 10 is oil we remain close and drill prospect 20, while if it is dry we move back to prospect 24. The 2-cells clustering shows, on the other side, some apparently less rational behavior like the suggestion of drilling prospect 24 no matter the outcome of prospect 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 behavior 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 prospect 14 and 19 are found dry we move to prospect 40, exploring a third new area.

6 Conclusion

What we have observed so far is that when there is a clear structure in the graphical model (parts of the network that are almost uncorrelated with other branches) it is extremely convenient to proceed with clustering algorithms such those proposed here. When on the other side, there is not a clear structure, even big clusters do not solve the problem of crossed-learning, and RHLA strategies perform better in terms of expected future revenues. It is worth noticing, though, that the main drawback of this method lies in the assumption that $\delta = 1$, thus removing that sequential effect that is so important in planning strategies. For this reason we believe that this method can help in approximating the continuation value, but is not really a viable alternative to RHLA if we are interested in sequential strategies.

The results show that without a natural clustering, it is difficult to assign correctly the nodes to each cluster. In a way, choosing $\{1, 2, 3, 4, 5, 6\}$ and $\{7, 8, 9, 10, 11, 12\}$ as the members of the first partition might be suboptimal. Further, it might be even less optimal to build the sequential bound in a random order. In this small case, though, there are $\begin{pmatrix} 12 \\ 6 \end{pmatrix}/2 = 462$ possible ordering of the first two clusters, and we have potentially to test all of the them before deciding which is the optimal partition.

The MRF results show that the both the clusters' size and shape are important when we are trying to build a sequential strategy, and prove that the learning process can be captured quite well even when we split the main problem in small sub-problems. In all the examples that we have tested, the proposed strategies and the proposed bounds perform much better than classical myopic/naive strategies.

We have not considered budgetary constraints in this 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. Neither did we consider the situation with imperfect information, i.e. when the nodes are only observed indirectly, and one can use this data to make better decisions about observing the node perfectly, or not. There are several interesting problems at the interface of statistical modeling and inference and operations research / decison 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 .

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.

H. 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 *0*.

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.

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, submitted .

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.

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.