Optimal Sequential Exploration: Bandits, Clairvoyants, and Wildcats

(Online Appendix)

David B. Brown and James E. Smith

Fuqua School of Business
Duke University
Durham, NC 27708-0120

dbbrown@duke.edu, jes9@duke.edu

First version: November 10, 2011
This version: February 7, 2013

B. Detailed Derivations and Results

B.1. Proof of Lemma 4.1: *Properties of Value Functions*

Proof. (i): Let π denote a policy for controlling a bandit superprocess and let Π denote the set of all such policies. (Here a policy π is a stationary policy that maps from the system state $\boldsymbol{x}$ to a choice of cluster i and action in $A(x_i)$ or else to "retire.")

For any policy π and initial state $\boldsymbol{x}$, we can decompose the value received into the sum of the discounted rewards $\tilde{r}(\pi, \boldsymbol{x})$ received prior to retirement and the discounted reward from retiring, $\delta^{\tilde{\tau}(\pi,\boldsymbol{x})}M$, where $\tilde{\tau}(\pi, \boldsymbol{x})$ is the time of retirement for policy π, starting in state $\boldsymbol{x}$. The expected value with a fixed policy π is then given as

$$\mathbb{E}\Big[\, \tilde{r}(\pi, \boldsymbol{x}) + \delta^{\tilde{\tau}(\pi,\boldsymbol{x})}M \,\Big|\, \boldsymbol{x} \,\Big] \ .$$

Notice that, for a fixed policy π and initial state $\boldsymbol{x}$, this is a nondecreasing (since $\delta^{\tilde{\tau}(\pi,\boldsymbol{x})} \geq 0$) and linear function of M.

We can express the value function $\Phi(\boldsymbol{x}, M)$ as

$$\Phi(\boldsymbol{x}, M) \quad = \quad \sup_{\pi\in\Pi} \mathbb{E}\Big[\, \tilde{r}(\pi, \boldsymbol{x}) + \delta^{\tilde{\tau}(\pi,\boldsymbol{x})}M \,\Big|\, \boldsymbol{x} \,\Big] .$$

Since $\Phi(\boldsymbol{x}, M)$ is the supremum of a set of increasing linear functions in M, $\Phi(\boldsymbol{x}, M)$ is increasing and convex in M. When the state and action spaces are finite, Π is a finite set, which implies that $\Phi(\boldsymbol{x}, M)$ is piecewise linear in M.

(ii): $\Phi'(\boldsymbol{x}, M)$ exists almost everywhere because $\Phi(\boldsymbol{x}, M)$ is convex in M. (ii)(a) follows directly from (i).

(ii)(b): Since $M < M^*(\boldsymbol{x})$, it cannot be optimal to retire immediately in state $\boldsymbol{x}$ when the retirement value is M: working exclusively on the cluster with largest Gittins index, for instance, yields an expected reward that is strictly larger than M. The result then follows by (ii).

(ii)(c): Consider some M where the derivative exists and let π^* be an optimal policy for this M. Now consider applying this policy to the problem with retirement value $M + \epsilon$; this yields a total value of

$$\Phi(\boldsymbol{x}, M) + \epsilon\mathbb{E}\Big[\, \delta^{\tilde{\tau}(\pi^*,\boldsymbol{x})} \,\Big|\, \boldsymbol{x} \,\Big] ,$$

where $\tilde{\tau}(\pi^*, \boldsymbol{x})$ is the retirement time for policy π^* and starting in state $\boldsymbol{x}$. Since π^* is feasible but not necessarily optimal given retirement value $M + \epsilon$, we know that

$$\Phi(\boldsymbol{x}, M + \epsilon) \quad \geq \quad \Phi(\boldsymbol{x}, M) + \epsilon\mathbb{E}\Big[\, \delta^{\tilde{\tau}(\pi^*,\boldsymbol{x})} \,\Big|\, \boldsymbol{x} \,\Big] .$$

Applying the same argument to the case with retirement value $M - \epsilon$, we get

$$\Phi(\boldsymbol{x}, M - \epsilon) \quad \geq \quad \Phi(\boldsymbol{x}, M) - \epsilon\mathbb{E}\Big[\, \delta^{\tilde{\tau}(\pi^*,\boldsymbol{x})} \,\Big|\, \boldsymbol{x} \,\Big] .$$

Combining these, we have

$$\frac{\Phi(\boldsymbol{x}, M) - \Phi(\boldsymbol{x}, M - \epsilon)}{\epsilon} \ \leq \ \mathbb{E}\big[\, \delta^{\tilde{\tau}(\pi^*,\boldsymbol{x})} \,\big|\, \boldsymbol{x} \,\big] \ \leq \ \frac{\Phi(\boldsymbol{x}, M + \epsilon) - \Phi(\boldsymbol{x}, M)}{\epsilon},$$

and taking $\epsilon \to 0$, the result follows. (This proof follows Bertsekas's (1995) proof of his Lemma 1.5.1.)

(iii): Increasing the retirement value from M to M' results in an increase of $M' - M$ in the reward of the policy that retires immediately in state $\boldsymbol{x}$. For any other policy that does not retire immediately, the change in the total expected reward in going from M to M' must be less than $M' - M$, as the retirement value is discounted. Since any such policy is not preferred to retiring immediately at M, it must remain optimal to retire immediately with M'.

These statements hold for the cluster-specific value functions because the cluster-specific case is a special case of a bandit superprocess with a single cluster. That $\phi_i'(x_i, M) = 1$ for $M > M_i^*(x_i)$ follows by (iii) and the definition of the Gittins index. □

B.2. Details on Bounds Based on Lagrangian Relaxations

To formalize the derivation of the Lagrangian relaxations, it is useful to augment the action sets $A_i(x_i)$ to sets $A_i^*(x_i)$ that include a "rest" action that pays no reward and results in no change of state. Assuming the state transitions are independent, we can then rewrite the bandit superprocess (2) with retirement value M=0 as

$$\begin{aligned} \Phi(\boldsymbol{x}, 0) = \max_{(a_1,\ldots,a_N)} \quad & \mathbb{E}\left[\sum_{i=1}^{N} \tilde{r}_i(x_i, a_i) + \delta \Phi(\tilde{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{a}), 0) \,\middle|\, \boldsymbol{x} \right] \\ \text{subject to} \quad & \sum_{i=1}^{N} \rho(a_i) \geq N - 1 \\ & a_i \in A_i^*(x_i) \quad \text{for } i = 1, \ldots, N \end{aligned} \tag{19}$$

where $\rho(a_i)$ is defined to be equal to one if a_i is the rest option and equal to zero otherwise. The first constraint here requires at least N-1 clusters to be resting or, equivalently, no more than one cluster to be active. The transitions $\tilde{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{a})$ and rewards $\tilde{r}_i(x_i, a_i)$ are defined in the obvious way with $\boldsymbol{a} = (a_1, \ldots, a_N)$; whenever the action a_i is to rest, the next state $\tilde{x}_i$ for cluster i equal to the current state x_i and the rewards $\tilde{r}_i(x_i, a_i)$ are zero.

Being a recursive equation, equation (19) imposes constraints for every state $\boldsymbol{x}$. Thus, when considering Lagrange multipliers associated with the first constraint in (19), we can take the Lagrange multipliers to be a function of the system state. Introducing a Lagrange multiplier function $\lambda(\boldsymbol{x}) \geq 0$, we can write the Lagrangian for (19) as

$$\begin{aligned} L(\boldsymbol{x}; \lambda(\cdot)) \equiv \max_{(a_1,\ldots,a_N)} \quad & \mathbb{E}\left[\sum_{i=1}^{N} \tilde{r}_i(x_i, a_i) + \delta L(\tilde{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{a}); \lambda(\cdot)) \,\middle|\, \boldsymbol{x} \right] + \lambda(\boldsymbol{x}) \left(\sum_{i=1}^{N} \rho(a_i) - (N-1) \right) \\ \text{subject to} \quad & a_i \in A_i^*(x_i) \quad \text{for } i = 1, \ldots, N. \end{aligned} \tag{20}$$

For any $\lambda(\boldsymbol{x}) \geq 0$, the Lagrangian provides an upper bound on the system-wide value function given in (19), that is, $L(\boldsymbol{x}; \lambda(\cdot)) \geq \Phi_0(\boldsymbol{x})$.

If we assume that the Lagrange multiplier function $\lambda(\boldsymbol{x})$ is a constant value λ ($\lambda \geq 0$), we can decompose the Lagrangian (20) into a series of cluster-specific problems that can be solved separately. Specifically, we can write $L(\boldsymbol{x}; \lambda)$ as

$$L(\boldsymbol{x}, \lambda) = \sum_{i=1}^{N} \ell_i(x_i, \lambda) - \frac{\lambda}{1-\delta}(N-1) \ , \tag{21}$$

where

$$\ell_i(x_i, \lambda) \equiv \max_{a_i \in A_i^*(x_i)} \mathbb{E}[\, \tilde{r}_i(x_i, a_i) + \lambda \rho(a_i) + \delta \ell_i(\tilde{x}_i(x_i, a_i)) \,|\, x_i \,] \ . \tag{22}$$

This result can be verified by substitution and was proven in Hawkins (2003; Theorem 1) and in Adelman and Mersereau (2008; Proposition 1).

The Lagrange multiplier λ in (21) and (22) can be interpreted as a subsidy paid to a cluster at rest. Note that in this bandit superprocess setting, if it is optimal to rest cluster i in one period in a state x_i, the cluster will rest forever as the state x_i will not change when resting. Thus we can interpret $M = \lambda/(1-\delta)$ as a retirement value and rewrite the cluster-specific functions $\ell_i(x_i, \lambda)$ equivalently in terms of the cluster-specific value functions with retirement value M, as defined in (3) as $\ell_i(x_i, M(1-\delta)) = \phi_i(x_i, M)$. We can then rewrite the system-wide Lagrangian as $L(\boldsymbol{x}, \lambda)$ as a function of $M = \lambda/(1-\delta)$ rather than a function of λ as:

$$\hat{L}(\boldsymbol{x}, M) \equiv L(\boldsymbol{x}, (1-\delta)M) = \sum_{i=1}^{N} \phi_i(x_i, M) - M(N-1). \tag{23}$$

We now show that in the North Sea example the best Lagrangian bound is given by taking M=0, in which case the Lagrangian $\hat{L}(\boldsymbol{x}, M)$ given by (11) reduces to the value given by allowing all of the clusters to be pursued simultaneously in the first period. To see this, note that, since $\hat{L}(\boldsymbol{x}, M)$ is piecewise-linear convex, if its derivative, $\hat{L}'(\boldsymbol{x}, M)$ is nonnegative for all M, then the minimum value must be obtained at M=0. (Recall that $M \geq 0$.) Calculating the derivative, this is equivalent to requiring

$$\sum_{i=1}^{N} \phi_i'(x_i, M) \geq (N-1). \tag{24}$$

for all M.

Recall from Lemma 4.1 that $\phi_i'(x_i, M)$ can be interpreted as $\mathbb{E}\big[\, \delta^{\tilde{\tau}_i(x_i,M)} \,\big|\, x_i \,\big]$ where $\tilde{\tau}_i(x_i, M)$ is the random time of retirement for cluster i when following the policy that is optimal for cluster i when viewed in isolation, given retirement value M and starting state x_i. In the North Sea example, regardless of the retirement value M, we know the stopping time $\tilde{\tau}_i(x_i, M)$ will certainly not exceed the number of targets in a cluster, which we denote n_i. Thus, for any M, we have

$$\sum_{i=1}^{N} \phi_i'(x_i, M) = \sum_{i=1}^{N} \mathbb{E}\Big[\, \delta^{\tilde{\tau}_i(x_i,M)} \,\Big|\, x_i \,\Big] \geq \sum_{i=1}^{N} \delta^{n_i} \ .$$

In our example, with a discount factor $\delta = 0.98$, for any choice of clusters, we find that $\sum_{i=1}^{N} \delta^{n_i}$ itself exceeds $N-1$, which implies (24) holds for all M; hence the optimal Lagrangian bound is given by taking $M = 0$. Specifically, in the case with one target per cluster, we have 25 clusters and $\sum_{i=1}^{N} \delta^{n_i} = 24.50$. In the case with clusters including all targets associated with each prospect, we have 13 clusters with $\sum_{i=1}^{N} \delta^{n_i} = 12.51$. For the cases of Figures 2 and 3, we have 6 and 4 clusters (respectively) and $\sum_{i=1}^{N} \delta^{n_i} = 5.52$ and 3.56. For the best Lagrangian bound to be something other than that given by simply considering the projects pursued in parallel immediately (i.e., with M=0), we would have to have many more targets or a much higher discount rate (i.e., lower discount factor δ).

B.3. Proof of Proposition A.1

Proof. (i) The proof is by induction. The initial basis matrix $\mathbf{B}_0 = \mathbf{I}$ is optimal for $M > M_0$ as M_0 is larger than the Gittins indices for all states. In the discussion before the proposition, we showed that if $\mathbf{B}_j$ is optimal for M_j, then it is optimal over the interval $[M_{j+1}, M_j]$. We need to show that the new basis is optimal at M_{j+1}. Let $\boldsymbol{\theta}_j$ denote a basic feasible solution to (18) with M_{j+1} and basis matrix $\mathbf{B}_j$; let $\bar{\boldsymbol{c}}$ and $\bar{\boldsymbol{d}}$ be the reduced costs for $\boldsymbol{c}$ and $\boldsymbol{d}$ with this basis, as defined in §A.2. The optimal objective function value is $(\boldsymbol{c} + M_{j+1}\boldsymbol{d})^{\mathsf{T}}\boldsymbol{\theta}_j$. Now consider a basic feasible solution $\boldsymbol{\theta}_{j+1}$ corresponding to basis $\mathbf{B}_{j+1}$ and let $\boldsymbol{\Delta} = \boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_j$. Using standard results from linear programming, we can express the change in objective function values when changing the basis as:

$$(\boldsymbol{c} + M_{j+1}\boldsymbol{d})^{\mathsf{T}}\boldsymbol{\Delta} = \sum_{k \in \mathcal{N}} (\bar{c}_k + M_{j+1}\bar{d}_k)\Delta_k \ , \tag{25}$$

where $\mathcal{N}$ denotes the set of non-basic indices for basis $\mathbf{B}_j$. Here, however, all of the non-zero components of Δ_k (representing those variables entering the basis) have reduced costs $(\bar{c}_k + M_{j+1}\bar{d}_k)$ equal to zero by definition of the set $\mathcal{I}$. Thus $(\boldsymbol{c} + M_{j+1}\boldsymbol{d})^{\mathsf{T}}\boldsymbol{\Delta} = 0$ and we can conclude that $\boldsymbol{\theta}_{j+1}$ and B_{j+1} are also optimal at M_{j+1}.

To see that optimal value function in the range $[M_{j+1}, M_j]$ is $\boldsymbol{\phi} = \boldsymbol{\lambda}_{cj} + M\boldsymbol{\lambda}_{dj}$ note that from (17), we have $\mathbf{B}_j^{\mathsf{T}}\boldsymbol{\phi} = \boldsymbol{c}_{\mathbf{B}} + M\boldsymbol{d}_{\mathbf{B}}$; this implies $\boldsymbol{\phi} = (\mathbf{B}_j^{\mathsf{T}})^{-1}\boldsymbol{c}_{\mathbf{B}} + M(\mathbf{B}_j^{\mathsf{T}})^{-1}\boldsymbol{d}_{\mathbf{B}} = \boldsymbol{\lambda}_{cj} + M\boldsymbol{\lambda}_{dj}$. Since $\phi_i(x_i, M) = \boldsymbol{\lambda}_{cj}(x_i) + M\boldsymbol{\lambda}_{dj}(x_i)$ for M in this range, it is clear that $\phi_i'(x_i, M) = \boldsymbol{\lambda}_{dj}(x_i)$.

(ii) The change in basis corresponds to a change in policy with the actions for the states in $\mathcal{I}$ changing at M_{i+1}. Since it is optimal to retire immediately for $M \geq M_0$ in any state, the first change in action for any given state corresponds to a change away from retirement (see Lemma 4.1(iii)); thus M_{j+1} for this first change is the Gittins index for this state. Note, however, that with bandit superprocesses not satisfying the Whittle condition, the optimal action for any given state may change multiple times.

(iii) Note that for any $M_j > 0$, since $M_j = \max\left\{-\bar{c}_k/\bar{d}_k : \bar{d}_k < 0\right\}$, we have $\bar{c}_k > 0$ for any variable entering the basis in the j^{th} iteration of the algorithm. Now consider the impact of the basis change on the value of the objective of (18) without considering the contribution of the retirement. Taking $\boldsymbol{\Delta} = \boldsymbol{\theta}_{j+1} - \boldsymbol{\theta}_j$ and reasoning as in (25), we have $\boldsymbol{c}^\mathsf{T}\boldsymbol{\Delta} = \sum_{k\in\mathcal{N}} \bar{c}_k \Delta_k$ where $\mathcal{N}$ is the set of non-basic indices. Since the entering values Δ_k are strictly positive (in fact, the Δ_k are all greater than one), we see that $\boldsymbol{c}^\mathsf{T}\boldsymbol{\theta}_j$ is strictly increasing with each iteration of the algorithm. Thus, the bases B_j will be distinct at each iteration and we do not have to worry about the algorithm cycling. Since there at most a finite number of different bases (or policies), the algorithm will terminate in a finite number of steps. □

B.4. Details on Calculating the Whittle Integral

To calculate the Whittle integral (5), we use the frontier algorithm to calculate the slopes and breakpoints for each cluster (i.e., $\phi_i'(x_i, M)$ for all $M \geq 0$) for a given state $\boldsymbol{x}$. Given these slopes and breakpoints, it is straightforward to calculate the Whittle integral $\hat{\Phi}(\boldsymbol{x}, 0)$.

First, we need to calculate the product of derivatives $\prod_{i=1}^N \phi_i'(x_i, m)$ for various values of m for this state. This product, like the cluster-specific derivatives, will be piecewise constant and increasing in m. The breakpoints for the product will be the union of breakpoints for the individual clusters; the slopes between these breakpoints can be calculated from the slopes for the individual clusters as follows. Assuming the slopes s_{ij} for cluster i are indexed in increasing order of the breakpoint values M_{ij}, we can define slope increments as $\hat{s}_{i1} = s_{i1}$ and, for $j > 1$, $\hat{s}_{ij} = s_{ij}/s_{i(j-1)}$. Using these increments, we can write:

$$\phi_i'(x_i, M) \quad = \quad \prod_{\{j:M_{ij}<M\}} \hat{s}_{ij}. \tag{26}$$

We can then represent the product of derivatives required for the Whittle integral calculation as

$$\prod_{i=1}^N \phi_i'(x_i, M) \quad = \quad \prod_{\{ij:M_{ij}<M\}} \hat{s}_{ij}. \tag{27}$$

Using this, we can combine the breakpoints and slope increments for all clusters and sort by their corresponding values of M_{ij} to obtain $\{M_k, \hat{s}_k\}_{k=1}^K$. The piecewise constant values of (27) are given by the cumulative product of these sorted slope increments, $s_k = \prod_{\kappa=1}^k \hat{s}_\kappa$. The breakpoints for the product will be the corresponding sorted values of M_k. The slope s_k applies from range M_k to M_{k+1}.

The integral in (5) is then given by weighing these piecewise constant values of this product of derivatives by the differences between adjacent breakpoints:

$$\hat{\Phi}(\boldsymbol{x}, 0) \equiv B - \int_{m=0}^B \prod_{i=1}^N \phi_i'(x_i, m)\ \mathrm{d}m \ = M_K - \sum_{k=1}^{K-1} s_k(M_{k+1} - M_k)\ .$$

Thus, once we have the slopes and breakpoints, these Whittle integrals can be computed quickly.

(Note that the formula for incremental slopes $\hat{s}_{ij}$ assumes that the slopes $s_{i(j-1)}$ are positive. If some slopes are zero, we can refine the algorithm to include only the values of m that have positive slopes for all clusters, as regions with zero slopes for one or more clusters contribute nothing to the Whittle integral.)

B.5. Proof of Proposition 5.1: *Clairvoyant Bound*

Proof. Suppose policy π for selecting clusters and actions is optimal for the original problem (1). Using this policy, we can unwind the dynamic program recursion of (1) and write the value as

$$V(\boldsymbol{x}^\circ) = \mathbb{E}\left[\sum_{t=0}^{\tilde{\tau}} \delta^t r_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi)) \,\middle|\, \boldsymbol{x}^\circ \right] \tag{28}$$

where $i_t(\pi)$, $a_{it}(\pi)$ and $\tilde{x}_{it}(\pi)$ denote the cluster, action selected, and state of the active cluster at time t under policy π, given starting state $\boldsymbol{x}^\circ$. $\tilde{\tau}$ is the (random) stopping time with this policy and starting state.

The clusters chosen and actions selected will all depend on the outcomes observed over time as well as the history of actions selected.

In the inner problem (14) for any $\hat{\boldsymbol{\omega}}$, the DM could choose clusters i to work on and actions a_i according to this same policy π that is optimal for the original problem (1). Assuming that this policy is followed, for a fixed outcome $\hat{\boldsymbol{\omega}}$, we can unwind the dynamic program recursion of (14) as we did in (28) above and find the value associated with following this policy:

$$V_c^{\pi}(\boldsymbol{x}^{\circ};\hat{\boldsymbol{\omega}}) = \mathbb{E}_{\hat{\boldsymbol{\omega}}}\left[\sum_{t=0}^{\tilde{\tau}} \delta^t r_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi)) \,\middle|\, \boldsymbol{x}^{\circ}\right] \tag{29}$$

where, as in (14), $\mathbb{E}_{\hat{\boldsymbol{\omega}}}$ denotes expectations taken with respect to this clairvoyant distribution $P_{\hat{\boldsymbol{\omega}}}$ for scenario $\hat{\boldsymbol{\omega}}$.

By the law of iterated expectations, we know that for any function f_i defined on x_i, $\mathbb{E}[\,\mathbb{E}_{\tilde{\boldsymbol{\omega}}}[\,f_i(\tilde{x}_{it}(\pi))\,|\,\boldsymbol{x}^{\circ}\,]\,] = \mathbb{E}[\,f_i(\tilde{x}_{it}(\pi))\,|\,\boldsymbol{x}^{\circ}\,]$. Applying this to (29), we find that $\mathbb{E}[\,V_c^{\pi}(\boldsymbol{x}^{\circ};\tilde{\boldsymbol{\omega}})\,]$ reduces to (28) and thus $\mathbb{E}[\,V_c^{\pi}(\boldsymbol{x}^{\circ};\tilde{\boldsymbol{\omega}})\,] = V(\boldsymbol{x}^{\circ})$. Thus, if we choose clusters actions in the dual problem according the optimal policy for the original problem (1), we obtain the same value. However, with the additional information in the inner problem (14), we can choose clusters and actions differently in each scenario $\hat{\boldsymbol{\omega}}$ and potentially improve upon $V_c^{\pi}(\boldsymbol{x}^{\circ};\hat{\boldsymbol{\omega}})$. Thus, if we choose clusters and actions optimally for each sampled $\tilde{\boldsymbol{\omega}}$, as we do in the clairvoyant bound $\mathbb{E}[\,V_c(\boldsymbol{x}^{\circ};\tilde{\boldsymbol{\omega}})\,]$, we can only improve upon $V(\boldsymbol{x}^{\circ})$. □

B.6. Control Variate Calculations

We consider two types of control variates in our calculations. For the clairvoyant bounds, we generate control variates by evaluating the policies used in static heuristic in the scenarios considered under the clairvoyant bound. Specifically,

- Before the simulation, for each cluster, we calculate the cluster-specific value function $\phi_i(x_i^{\circ})$ in its initial state and the corresponding optimal policy π_i, using the marginal distributions $P(\omega_i|\boldsymbol{x}^{\circ})$ for that cluster; we assume zero retirement value. This policy π_i is also used to specify actions in the static heuristic.
- Then, for each sample $\hat{\boldsymbol{\omega}}$ of the simulation, we calculate the value of this cluster with this policy π_i, using the marginal distributions used with the clairvoyant bounds, $P(\omega_i|\hat{\boldsymbol{\omega}}_{\bar{i}}, \boldsymbol{x}^{\circ})$. We denote this value by $\phi_i^{\pi_i}(x_i^{\circ};\hat{\boldsymbol{\omega}})$. (Here retirement decisions and actions are determined by π_i.)
- Since the policy is fixed, $\mathbb{E}[\,\phi_i^{\pi_i}(x_i^{\circ};\tilde{\boldsymbol{\omega}})\,] = \phi_i(x_i^{\circ})$. Thus, after the simulation, we can use the differences $\phi_i^{\pi_i}(x_i^{\circ};\tilde{\boldsymbol{\omega}}) - \phi_i(x_i^{\circ})$ for all clusters as a vector of control variates for the clairvoyant bounds. We use a standard multiple-regression approach to control variates (see, e.g., Glasserman (2004)) to correct the value estimates.

Because the policies π_i and the distributions involved are already used in the static heuristic and clairvoyant bounds, there is little additional work required in these control variate calculations.

For the heuristic policies, we apply the control variates discussed above in addition to another control variate constructed using the "approximating martingale-process method" of Henderson and Glynn (2002). This method is based on using an easy-to-evaluate function that approximates the reward-to-go at time t and taking expectations to form a martingale that can be used as a control variate. We will use the system-wide value function $\Phi_{\alpha}^{\pi}(\boldsymbol{x})$ under the multiarmed bandit-based approximation (13) to construct this martingale approximation. Since (13) is a multiarmed bandit, we can calculate $\Phi_{\alpha}^{\pi}(\boldsymbol{x})$ exactly by evaluating the corresponding Whittle integral.

If we denote the cluster and action chosen by the heuristic as $i_t(\pi)$ and $a_{it}(\pi)$, respectively, and let $\tilde{x}_{it}(\pi)$ and $\tilde{\boldsymbol{x}}_t(\pi)$ denote the state of cluster i and the system at time t under policy π, we can approximate the reward-to-go at time t given outcome $\hat{\boldsymbol{\omega}}$ as

$$\tilde{r}_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi), \hat{\omega}_{i_t(\pi)}) + \delta\Phi_{\alpha}^{\pi}(\tilde{\boldsymbol{x}}(\boldsymbol{x}_t(\pi), a_{it}(\pi), \hat{\boldsymbol{\omega}}))\ .$$

The expected value of this reward-to-go approximation given state $\boldsymbol{x}$ is

$$\mathbb{E}\big[\,\tilde{r}_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi)) + \delta\Phi_{\alpha}^{\pi}(\tilde{\boldsymbol{x}}(\boldsymbol{x}_t(\pi), a_{it}(\pi)))\,\big|\,\boldsymbol{x}_t(\pi)\,\big]\ ,$$

where expectations are calculated using the full model, i.e., using $P(\omega_i|\boldsymbol{x})$. In the static approach for the bandit approximation, we keep π and Φ^π_α fixed in the calculation of these expectations. In the sequential approach, we update π and Φ^π_α in each period based on the results observed in early periods.

In each period, the differences between the approximate reward-to-go and its expectation,

$$\begin{aligned}&\tilde{r}_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi), \hat{\omega}_{i_t(\pi)}) + \delta\Phi^\pi_\alpha(\tilde{\boldsymbol{x}}(\boldsymbol{x}_t(\pi), a_{it}(\pi), \hat{\boldsymbol{\omega}}))\\ &- \mathbb{E}\big[\, \tilde{r}_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi)) + \delta\Phi^\pi_\alpha(\tilde{\boldsymbol{x}}(\boldsymbol{x}_t(\pi), a_{it}(\pi))) \,\big|\, \boldsymbol{x}_t(\pi) \,\big],\end{aligned}$$

has mean zero, so these terms define a martingale. Summing across all times up to the stopping time $\tilde{\tau}$ for the heuristic policy in this sample, the value

$$\begin{aligned}&\sum_{t=0}^{\tilde{\tau}} \big(\tilde{r}_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi), \hat{\omega}_{i_t(\pi)}) + \delta\Phi^\pi_\alpha(\tilde{\boldsymbol{x}}(\boldsymbol{x}_t(\pi), a_{it}(\pi), \hat{\boldsymbol{\omega}}))\\ &- \mathbb{E}\big[\, \tilde{r}_{i_t(\pi)}(\tilde{x}_{it}(\pi), a_{it}(\pi)) + \delta\Phi^\pi_\alpha(\tilde{\boldsymbol{x}}(\boldsymbol{x}_t(\pi), a_{it}(\pi))) \,\big|\, \boldsymbol{x}_t(\pi) \,\big]\big)\end{aligned}$$

has mean zero and thus provides a valid control variate.

If $\Phi^\pi_\alpha(\boldsymbol{x})$ is a good approximation to actual reward-to-go under the heuristic, then we would expect this control variate to be highly correlated with the discounted rewards generated by the heuristic and, hence, to reduce the sample variation in the estimated values. The key issue in this approximation is that the bandit-based approximation (13) underlying $\Phi^\pi_\alpha(\boldsymbol{x})$ assumes the clusters are independent (i.e., it ignores cross cluster learning) whereas the simulations and the expectations involved in constructing the martingale include all dependence in the model. Intuitively, if "most" of the dependence in the model is captured within the clusters, this control variate will be very effective.

Most of the terms involved in these martingale control variates are calculated already, e.g., when running the frontier algorithm or updating the probability distributions over time, as in the sequential heuristic. The exception is that the expectations in the control variate calculations require probabilities for next-period states that are not needed to determine the value generated by the static heuristic, but are needed for the control variate calculation. Our experiments suggest that the improvement in accuracy provided by the control variate calculation for the static heuristic is worth the extra computational effort required: i.e., we achieve better accuracy for a given run time using the control variate than we would by simply running more trials of the simulation.

B.7. Details on Choosing Clusters for the North Sea Example

Table B1 provides the detailed results for individual clusters that we reviewed when selecting clusters in the North Sea example. We followed the general procedure outlined in §5.3:

- For each cluster, we calculated the cluster-specific value $\phi_i(x^0_i)$ using the marginal probability distribution for that cluster. This represents the value (in the initial state) for that cluster considered in isolation. These values can be calculated exactly (without simulation) using the frontier algorithm or a policy method.
- Then, in the simulation study, we calculated the value of the cluster given perfect information about the outcomes for all other clusters, $\phi_{ci}(x^0_i; \hat{\omega}_{\bar{i}})$, in each simulated scenario. This value is calculated as part of the frontier algorithm used for calculating clairvoyant Whittle or Lagrangian bounds and requires no additional work. The average of these values, $\mathbb{E}\big[\, \phi_{ci}(x^0_i; \hat{\omega}_{\bar{i}}) \,\big]$, represents the value of the cluster with perfect information about all other clusters. As these values are estimated using simulation, we report mean standard errors as well as means in Table B1.
- The difference between the two figures, $\mathbb{E}\big[\, \phi_{ci}(x^0_i; \hat{\omega}_{\bar{i}}) \,\big] - \phi_i(x^0_i)$, represents the expected value of information (EVPI) provided by other clusters for cluster i.

In choosing clusters, we identified the clusters that had the largest EVPIs and then sought to combine them with neighboring clusters, which are presumably providing the valuable information. To simplify the discussion in the paper, we used the same sets of clusters for the cases with and without kitchen uncertainty. Here we will focus on the results for the case without kitchen uncertainty, shown in the left side of Table B1.

First, we considered the case where each target was modeled as its own cluster. This case was considered for primarily for expository reasons; we expected targets to benefit greatly from information about other

targets associated with the same prospect. As discussed in §2, if oil or gas is found at any target associated with a prospect, this proves that there is oil or gas at the prospect and increases the probability of finding oil at gas at other targets associated with the same prospect. Conversely, dry results at one target may lead to a higher probability of dry wells at targets associated with the same prospect. In part (a) of Table B1, we see that many targets have large EVPIs and are thus learning a great deal from the results of other targets. It seemed natural to next combine all of targets associated with each prospect into the same cluster, leading to a model with one cluster per prospect.

When considering the results with clusters corresponding to prospects (shown in part (b) of Table B1), we observed that prospects 7 and 11 were both benefiting significantly from the information from other clusters, with EVPIs of \$450.8M and \$603.4M, respectively. These clusters are adjacent in the network and, as these clusters each contain a single target, they could easily be combined. Prospects 8 and 12 are also near prospects 7 and 11 in the network and though they were not benefitting as much from the additional information, we conjectured that these prospects may be contributing valuable information to prospects 7 and 11. In the next study, we combined prospects 7, 8, 11, and 12 into a single cluster with 5 targets; this combined cluster is cluster 6 in the case with medium clusters shown in Figure 2. Similar logic led us to combine clusters 5 and 9 and also clusters 10 and 13 into clusters 3 and 5 in Figure 2.

In the results with medium clusters (shown in part (c) of Table B1), we see that then new cluster 6 (containing prospect 7, 8, 11, and 12) is now benefitting little from other clusters: the EVPI for the combined cluster is now only \$2.2M. Clusters 4 and 5 are however still benefitting a fair amount from information from others, with EVPIs of \$104.1M and \$130.1M. In the next run, we combined these two clusters into a large cluster containing 9 targets; this is cluster 3 in Figure 3. (We also dropped target 6C, as discussed in §6.3.) In part (d) of Table B1, we see that this large cluster is learning much less, with an EVPI of only \$14.0M. Because larger clusters would be time consuming to evaluate and the differences between heuristic values and clairvoyant bounds were quite small, we chose to stop at this point.

B.8. Formal Justification for Dropping Targets 6C and 8A

We can prove that it is not optimal to drill Targets 6C and 8A in the following way. If we condition on the corresponding parent prospect having oil or gas, we can directly compute the most favorable possible expected reward at each target. For target 6C, the most favorable situation is for its parent prospect, P6, to have oil, which yields an expected reward of -\$874M. For target 8A, the most favorable case is when its parent prospect P8 contains either oil or gas, which leads to an expected reward at 8A of -\$828M.

We can compare these best-case expected rewards (which are negative) to the possible information benefit of drilling at either target: this value of information benefit is bounded by the difference in the clairvoyant Whittle bound and the sequential heuristic bound with any definition of clusters that has 6C and 8A isolated in individual clusters. We ran a simulation similar to the case with clusters as defined in Figure 2, but with 6C and 8A in clusters of their own, and found that the gap between the bounds was substantially less than \$828M, with or without kitchen uncertainty. Since the most favorable value of drilling at these sites is a substantial cost that cannot be made up by a commensurate gain in information value, we conclude that neither target will ever be drilled in an optimal policy.

B.9. Details on Bounds with Fixed First Actions

We know that an optimal policy must choose one action in the first period. However, in the clairvoyant bound calculations, the clairvoyant values in different simulated scenarios may be derived from different choices of first-period actions, depending on the outcomes of events that would not (yet) be known by a non-clairvoyant DM. Indeed, it precisely this kind of flexibility that makes clairvoyance valuable.

We can restrict this flexibility and improve the bounds by imposing constraints on the choice of first-actions and calculating first-action constrained bounds. Consider a particular action a_i for cluster i and a particular simulated scenario $\hat{\boldsymbol{\omega}}$. We can calculate a constrained clairvoyant Whittle value (analogous to the unconstrained Whittle integral bound $\hat{\Phi}_c(\boldsymbol{x};\hat{\boldsymbol{\omega}})$, as defined in §5.2) for this scenario as:

$$\hat{\Phi}_c^{a_i}(\boldsymbol{x};\hat{\boldsymbol{\omega}}) \quad \equiv \quad \mathbb{E}_{\hat{\boldsymbol{\omega}}}\Big[\, \tilde{r}_i(x_i,a_i) + \delta\hat{\Phi}_c(\tilde{\boldsymbol{x}}(\boldsymbol{x},a_i);\hat{\boldsymbol{\omega}}) \,\Big|\, x_i \,\Big] \tag{30}$$

where, as in (14), $\mathbb{E}_{\hat{\boldsymbol{\omega}}}$ denotes expectations calculated using the clairvoyant distribution distribution $P_{\hat{\boldsymbol{\omega}}}$ for scenario $\hat{\boldsymbol{\omega}}$.

Table B1: Cluster-specific results (All values are $M)

a) Target-level clusters

	Without Kitchen Uncertainty				With Kitchen Uncertainty			
Target	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI
1A	1755.9	1760.8	(0.2)	5.0	1057.2	1378.6	(9.2)	321.4
2A	0	1.1	(0.6)	1.1	0	1.4	(0.8)	1.4
3A	0	0	(0)	0	0	2.0	(2.0)	2.0
4A	0	13.3	(1.0)	13.3	0	48.6	(3.0)	48.6
4B	729.1	764.8	(4.1)	35.7	546.3	692.7	(6.2)	146.4
5A	533.9	667.4	(0.5)	133.5	143.6	549.2	(0.7)	405.6
5B	844.4	1166.1	(1.2)	321.7	0	962.0	(29.9)	962.0
5C	2107.3	2294.6	(0.9)	187.3	1315.7	1883.6	(1.4)	567.9
6A	1965.4	2165.5	(8.5)	200.1	969.5	1503.7	(14.7)	534.2
6B	1747.0	1849.0	(5.2)	102.0	1018.7	1263.7	(9.2)	244.9
6C	0	0	(0)	0	0	0	(0)	0
7A	122.7	573.5	(0.2)	450.8	0	466.1	(17.0)	466.1
8A	0	0	(0)	0	0	0	(0)	0
9A	0	122.5	(2.2)	122.5	0	82.0	(2.8)	82.0
9B	319.2	525.8	(0.3)	206.7	0	424.2	(13.5)	424.2
9C	172.1	526.8	(0.4)	354.7	0	423.7	(13.5)	423.7
10A	0	18.1	(3.0)	18.1	0	83.5	(5.6)	83.5
10B	5697.8	5805.3	(12.7)	107.5	4862.8	5110.4	(15.3)	247.7
10C	1284.8	1863.8	(1.6)	579.0	864.9	1642.2	(1.9)	777.3
11A	0	603.4	(14.1)	603.4	0	429.7	(16.1)	429.7
12A	335.9	1058.7	(8.1)	722.8	211.4	1088.9	(8.7)	877.5
12B	783.4	1510.9	(9.5)	727.5	112.5	1130.2	(11.5)	1017.7
13A	806.1	1063.7	(0.9)	257.6	516.6	993.6	(1.0)	477.0
13B	2051.8	2239.3	(9.8)	187.4	1356.9	1724.6	(12.8)	367.8
13C	628.7	841.1	(10.8)	212.3	143.8	552.3	(13.2)	408.5

b) Prospect-level clusters

	Without Kitchen Uncertainty				With Kitchen Uncertainty			
Prospect	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI
1	1755.9	1760.8	(0.2)	5.0	1057.2	1378.6	(9.2)	321.4
2	0	1.1	(0.6)	1.1	0	1.4	(0.8)	1.4
3	0	0	(0)	0	0	2.0	(2.0)	2.0
4	741.3	742.3	(0.4)	1.0	595.2	695.9	(2.5)	100.7
5	3684.5	4039.8	(6.0)	355.3	2552.0	3328.9	(7.9)	776.9
6	3749.7	3853.8	(4.3)	104.1	2325.1	2574.5	(8.2)	249.4
7	122.7	573.5	(0.2)	450.8	0	466.1	(17.0)	466.1
8	0	0	(0)	0	0	0	(0)	0
9	966.4	1123.6	(2.3)	157.1	412.9	898.4	(3.5)	485.5
10	7500.5	7610.3	(4.5)	109.7	6451.5	6755.8	(6.4)	304.3
11	0	603.4	(14.1)	603.4	0	429.7	(16.1)	429.7
12	1821.6	1911.8	(6.5)	90.2	1312.8	1578.5	(12.4)	265.6
13	3896.8	3918.2	(7.3)	21.4	2863.2	2991.3	(14.3)	128.1

c) Medium clusters (as in Figure 2)

	Without Kitchen Uncertainty				With Kitchen Uncertainty			
Cluster	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI
1	1755.9	1762.9	(0.7)	7.1	1057.2	1376.0	(4.3)	318.8
2	741.3	742.3	(0.4)	1.0	595.2	695.9	(2.5)	100.7
3	4867.3	4877.2	(0.5)	9.9	3625.6	3693.6	(3.6)	67.9
4	3749.7	3853.8	(4.3)	104.1	2325.1	2574.5	(8.2)	249.4
5	11263.0	11393.1	(6.9)	130.1	9240.9	9651.2	(14.9)	410.3
6	2509.4	2511.6	(0.0)	2.2	1602.8	1602.8	(0)	0

d) Large clusters (as in Figure 3)

	Without Kitchen Uncertainty				With Kitchen Uncertainty			
Cluster	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI	$\phi_i(x_i^\circ)$	$E[\phi_{ci}(x_i^\circ; \omega)]$	(mse)	EVPI
1	5541.2	5542.1	(0.1)	0.9	4241.0	4247.8	(0.9)	6.8
2	0	2.0	(0.8)	2.0	0	3.3	(1.1)	3.3
3	16409.0	16423.0	(1.3)	14.0	12747.6	12844.2	(6.4)	96.7
4	2509.4	2511.6	(0.0)	2.2	1602.8	1602.8	(0)	0

Since the Whittle integral provides an upper bound on the state-contingent continuation values, the average of these values, $\mathbb{E}\Big[\hat{\Phi}_c^{a_i}(\boldsymbol{x}^\circ;\tilde{\boldsymbol{\omega}})\Big]$, provides an upper bound on the value with the constraint of starting with action a_i first. Because an optimal policy has to select some action first, we know that the maximum of the bounds $\mathbb{E}\Big[\hat{\Phi}_c^{a_i}(\boldsymbol{x}^\circ;\tilde{\boldsymbol{\omega}})\Big]$ over first-period actions a_i provides a valid upper bound for the optimal sequential exploration problem. Moreover, this bound cannot be any worse than the unconstrained clairvoyant Whittle bound.

If we are calculating unconstrained clairvoyant Whittle bounds, these first-action constrained bounds can be calculated in the same simulation with relatively little additional work. One pass of the frontier algorithm generates all of the break points and slopes required to calculate the Whittle integral in any given state: we need only record the appropriate values for the possible next-period states given action a_i and combine them to find the Whittle integral for these next-period states. The other information required to evaluate $\hat{\Phi}_c^{a_i}(\boldsymbol{x}^\circ;\hat{\boldsymbol{\omega}})$ (e.g., the appropriate transition probabilities) are used in the unconstrained dual bounds as well.

Table B2 shows the results for these first-action-fixed clairvoyant bounds in the cases with large clusters, with the targets sorted by the value of the bound. The lines indicate where the estimated value of the static heuristic falls in this list. In the case without kitchen uncertainty, the expected value of the static heuristic is \$23,150M (with a mean standard error of \$5M); with kitchen uncertainty, the expected value is \$17,717M (with a mean standard error of \$19M). Setting aside sampling error, all targets that lie below these values can be ruled out as optimal first actions because the bounds show that policies that start with that target cannot perform as well as the static heuristic.

We could extend this idea of constraining the first action to constraining the first two periods by fixing the choice of actions in the first period and the state-contingent choice of actions in the second period. To generate an upper bound, we would have to consider all possible second choices of actions for each outcome of the first action chosen, i.e., all possible policies for the first two periods. In the North Sea example, given that we can rule certain first-period actions, we need only consider state-contingent second-period actions for first-period actions that have not been ruled out in the earlier analysis.

This same idea could be applied at deeper levels – e.g., constraining the policies used in the first three or four periods – in a similar manner. However, the number of policies that must be considered would likely grow quickly with increasing depth.

Table B2: Clairvoyant Whittle Bounds with Fixed First Actions
(Clusters as in Figure 3 case)

Without Kitchen Uncertainty			With Kitchen Uncertainty		
Target	Mean ($M)	MSE ($M)	Target	Mean ($M)	MSE ($M)
10B	*23,248*	2	10B	*17,894*	6
13B	*23,152*	1	6B	*17,782*	6
6B	23,142	1	13B	*17,729*	2
6A	23,136	1	6A	17,706	6
1A	23,132	1	12A	17,688	6
5C	23,086	2	4B	17,667	6
12B	23,002	2	9B	17,633	6
4B	22,986	2	9A	17,628	6
12A	22,986	1	5A	17,627	6
9B	22,975	1	1A	17,594	7
13A	22,964	1	12B	17,592	6
5A	22,943	2	13A	17,559	2
9A	22,894	1	5C	17,503	6
7A	22,866	2	13C	17,334	3
9C	22,827	2	7A	17,321	6
5B	22,824	2	9C	17,213	6
13C	22,820	2	10C	17,205	6
10C	22,631	2	2A	17,168	8
2A	22,539	3	5B	17,042	6
4A	22,477	2	4A	17,018	6
11A	22,422	2	11A	16,416	6
3A	21,771	9	3A	16,235	14
10A	21,387	2	10A	16,118	6