

New Regression Monte Carlo Methods for High-dimensional Real Options Problems in Minerals industry

N. Langrené^a, T. Tarnopolskaya^b, W. Chen^b, Z. Zhu^a, and M. Cooksey^a

^a*CSIRO, Clayton, VIC, 3168, Australia*
^b*CSIRO, North Ryde, NSW, 2113, Australia*
Email: nicolas.langrene@csiro.au

Abstract Mining operations are affected by significant uncertainty in commodity prices, combined with geological uncertainties (both in quantity and quality of the available reserves). Technical difficulties and costs associated with ore extraction together with a highly uncertain environment present significant risks for profitability of mineral projects. Optimising operating strategies in response to changing market conditions and information about the available reserves is crucial for project profitability in the face of uncertainty.

A natural resource extraction problem can be viewed as a stochastic optimal control (real options) problem, with extraction rate representing a control variable. In a finite horizon, finite reserve setting, an additional complexity arises from the need to consider a large number of feasible remaining reserve levels, which significantly increases the computational complexity of the algorithms.

Extraction of a natural resource problems have attracted the attention of researchers in the fields of real options and stochastic optimal control since the 1980s. However, there is still no computational framework available that would allow realistic high-dimensional real options problems in minerals industry to be solved.

Over the last decade, the approach based on value function approximation via basis functions has attracted significant attention from financial applications, and has given rise to a class of methods known as regression Monte Carlo methods. Regression Monte Carlo is a very versatile simulation-based technique. It can deal with a rich description of the mining problem, and very elaborate models for the risk factors.

In this paper, we propose to combine several crucial improvements to make the regression Monte Carlo method practical for multi-dimensional models:

- 1) Firstly, we avoid the discretisation of reserve level by using the control randomization technique. First, the reserve is replaced by a dummy random factor during the forward loop. Then, this variable is included into the regression factors during the backward loop, and optimised. Randomization also allows dealing with geological uncertainties in the estimated reserve.
- 2) Then, to avoid the full storage of the sample paths, we implement a memory reduction method. The idea is to store the seeds of the random number generator during the forward loop, in order to reproduce the paths exactly during the backward loop. This drastically reduces memory consumption.
- 3) Finally, to solve once and for all the problem of choice of regression basis, we perform non-parametric adaptive local regressions, which automatically adapt to the data and the function to regress. Its numerical efficiency is ensured by a novel fast implementation of the method.

We explain how these efficient implementation techniques allow us to tackle a stylized mineral extraction problem under both price and geological uncertainties. One key advantage of the proposed improvements is that they are easily extendable to higher dimensions and make it possible to tackle realistic multi-dimensional real option problems. For the mining industry, this means better estimates for the value of a mine, with geological and price uncertainties taken into account. Beyond that, it means better dynamic strategies for mine operation, with explicit rules on how to deal with changing circumstances.

Keywords: *Real option, stochastic control, Monte Carlo, control randomization, memory reduction*

1 INTRODUCTION

It is broadly acknowledged that mining operations are highly affected by risk and uncertainty, with commodity price and geological uncertainty (in both quantity and quality of the estimated reserve) being the most relevant risk factors that attract considerable attention in the research literature (see Castillo and Dimitrakopoulos (2014), Ndiaye and Armstrong (2013) and the references therein). The main obstacle to include multiple uncertainties into real options analysis is that existing real options/stochastic control methodologies are generally not capable of dealing with more than three stochastic risk factors.

Over the last decade, the approach based on value function approximation via basis functions, introduced by Longstaff and Schwartz (2001) under the name Least-Squares Monte Carlo (LSMC), has attracted significant attention for financial applications. LSMC is a very versatile simulation-based technique which is easily extendable to higher dimensions. It has been extended to more general regression methods (Regression Monte Carlo) and to more general stochastic control problems, such as optimal switching problems, which include minerals real options problems when a discrete grid is used to describe the possible reserve levels (see, e.g., Dimitrakopoulos and Abdel Sabour (2007), Bao *et al.* (2013)). However, a few bottlenecks arise for high-dimensional optimal switching problems: (a) computational complexity due to the number of switching levels (e.g. reserve levels); (b) memory complexity; and (c) difficulty of choosing an appropriate regression basis.

In this paper, we propose to combine several crucial improvements to make the regression Monte Carlo method practical for multi-dimensional models in the minerals applications: control randomization, memory reduction and fast adaptive local regression. We show that the regression Monte Carlo algorithm with the above improvements can easily accommodate realistic models of stochastic commodity price dynamics with jump, in addition to geological uncertainties (in both the quality and quantity of estimated reserve), and can deal with many more risk factors if needed.

The paper is organized as follows. Section 2 provides the problem formulation. It shows how taking a few additional risk factors into account allows us to account for geological uncertainties in a stylized fashion. Section 3 describes a few methods to make high dimensional problems manageable by regression Monte Carlo, namely control randomization, memory reduction and fast local polynomial regression. Section 4 provides a few numerical results and a discussion of the potential paths to explore in the future. Finally, Section 5 concludes this paper.

2 PROBLEM FORMULATION

We study how extraction decisions for a given mine should be made to maximise expected financial gains over the finite planning horizon, with flexibility to operate, temporarily close, restart or permanently abandon the mine. We extend the model used in Chen *et al.* (2015) to include a more realistic mean reversion stochastic process with jump to describe the ore price S . We also consider the estimated reserve Q and the extraction rate q as additional stochastic risk factors. We model the estimated reserve Q as a geometric Brownian motion with jump, in order to account for both small random adjustments in the estimated remaining reserve and sharp changes due to unexpected new discovery or unforeseen early depletion. In order to account for the uncertainty related to ore grade, we consider the extraction rate as a stochastic variable, thus accommodating for uncertain content of metal in the ore. Together, Q and q account for geological uncertainties related to both quantity and quality of ore. We model the dynamics of risk factors as follows:

$$dS_t = \kappa_S (\bar{S} - S_t) dt + \sigma_S S_t dW_S(t) + (J_S - 1) S_t dP_S(t), \quad t \in [0, T] \quad (2.1)$$

$$dq_t = \kappa_q (\bar{q} - q_t) dt + \sigma_q q_t dW_q(t), \quad t \in [0, T] \quad (2.2)$$

$$dQ_t = -q_t \mathbb{1}_{\{\alpha_t=o\}} dt + \sigma_Q \mathbb{1}_{\{\alpha_t=o\}} Q_t dW_Q(t) + (J_Q - 1) \mathbb{1}_{\{\alpha_t=o\}} Q_t dP_Q(t), \quad t \in [0, T \wedge \tau_d] \quad (2.3)$$

where $W = (W_S(t), W_q(t), W_Q(t))_{t \in [0, T]}$ is a 3-dimensional Brownian motion, σ_S , σ_q and σ_Q are constant volatilities, \bar{S} and \bar{q} are constant average levels, $\tau_d := \inf \{t \geq 0; Q_t = 0\}$ (depletion) and $\mathbb{1}_{\{\cdot\}}$ is the indicator function. $\alpha = (\alpha_t)_{0 \leq t \leq T}$ is the dynamic control, or policy which takes values in the set of operating regimes $\mathbb{Z} = \{o, c, a\}$, where o , c and a denote an open (operating), a temporarily closed and an abandoned mine respectively. The control α is a function of t , S , q and Q , as decisions take all the available information into account. $J_A = e^{\phi_A}$, $\phi_A \sim \mathcal{N}(-\sigma_{J_A}^2/2, \sigma_{J_A}^2)$, $A \in \{S, Q\}$ are jump components which occur at random Poisson times

$$dP_A(t) = \begin{cases} 1 & \text{with prob } \lambda_A dt \\ 0 & \text{with prob } 1 - \lambda_A dt. \end{cases}$$

Let $\theta = [q, Q, i]$ be the state vector and $\theta_t = [q_t, Q_t, \alpha_t]$ be the state vector at time t . The value function of the problem is then defined by:

$$V(t, S, \theta) = \sup_{\alpha \in \mathcal{A}} \mathbb{E} \left[\int_t^{T \wedge \tau_a} e^{-\tilde{r}(s-t)} \Pi(s, S_s, \alpha_s) ds - \sum_{t \leq \tau_n \leq T \wedge \tau_a} e^{-\tilde{r}(\tau_n-t)} k(\tau_n, \alpha_{\tau_n^-}, \alpha_{\tau_n}) | (S_t, \theta_t) = (S, \theta) \right] \quad (2.4)$$

where $\tau_a := \inf \{t \geq 0; \alpha_t = a\}$ (abandonment). The notations used to define the value function are the same as in Chen et al. (2015): $\tilde{r} = r + \lambda$ is the discount rate, where r is the risk-free rate and λ is the property tax rate; $\Pi(s, S_s, \alpha_s)$ is the instantaneous cash-flow at time s when the metal price is equal to S_s and the operating regime is equal to $\alpha_s \in \mathbb{Z}$; $k(\tau_n, \alpha_{\tau_n^-}, \alpha_{\tau_n})$ is the switching cost at stopping time of switching τ_n from regime $\alpha_{\tau_n^-}$ to α_{τ_n} .

3 NUMERICAL METHODS

The problem formulation can be seen as an extension of Chen et al. (2015), where only two state variables were considered: a stochastic exogenous state variable S and a deterministic endogenous state variable Q . Here we have three main risk factors: S is a stochastic exogenous state variable that features jumps and mean-reversion; q is now also a stochastic exogenous state variable. Finally, Q is a stochastic endogenous state variable with jumps. Dealing with dynamics with jumps is not a problem when using the regression Monte Carlo approach. However, dealing with a stochastic endogenous state variable is a major difficulty, as the reserve discretization from Chen et al. (2015) cannot be easily used. Our solution is to replace it with the control randomization technique, described in Subsection 3.1. This technique requires the use of an additional dummy stochastic factor, making our problem four-dimensional. Consequently, the memory gains provided by the memory reduction method described in Subsection 3.2 can become substantial. Moreover, guessing a suitable regression basis may become difficult, which makes the fast adaptive local regression described in Subsection 3.3 very useful.

3.1 Control randomization

The control randomization technique, pioneered in Kharroubi et al. (2014), makes it possible to solve stochastic control problems with endogenous risk factors by a probabilistic algorithm (Regression Monte Carlo). A risk factor is endogenous when its dynamics depend on the control, which is precisely the case here for the estimated reserve Q (2.3). More generally, let X be a (possibly multivariate) endogenous risk factor $dX_t^\alpha = b(t, X_t^\alpha, \alpha_t) dt + \sigma(t, X_t^\alpha, \alpha_t) dW_t$, $X_0^\alpha = x_0$, where α is the control. Suppose that one wants to solve the following stochastic control problem:

$$v(t, x) = \sup_{\alpha} \mathbb{E} \left[\int_t^T f(s, X_s^\alpha, \alpha_s) ds + g(T, X_T^\alpha) | X_t^\alpha = x \right] \quad (3.1)$$

To do so, one can still use a Monte Carlo algorithm. The key ingredient is control randomization, ie. the state variable is first simulated with a dummy random control $\tilde{\alpha}$:

$$d\tilde{X}_t = b(t, \tilde{X}_t, \tilde{\alpha}_t) dt + \sigma(t, \tilde{X}_t, \tilde{\alpha}_t) dW_t, \quad \tilde{X}_0 = x_0 \quad (3.2)$$

Then, one can implement the backward induction described in Algorithm 1. In the spirit of Longstaff and Schwartz (2001), Algorithm 1 regressed realized cash-flows instead of value function, which requires path recomputation when risk factors are endogenous.

3.2 Memory reduction

The Regression Monte Carlo algorithm 1 contains a forward loop and a backward loop. During the forward loop, all the (randomized) risk factors are simulated until maturity T . Then, this sample of paths is used during the backward loop. The usual approach is to store all the paths after the forward loop, for later use during the backward loop. This requires a substantial $\mathcal{O}(N \times M)$ in memory size, where N is the number of time steps and M is the number of Monte Carlo paths. Fortunately, this can be reduced to $\mathcal{O}(N + M)$ thanks to the memory reduction method proposed in Aïd et al. (2014) for the Euler-Maruyama scheme and in Hu and Li (2014) for the Milstein scheme.

Algorithm 1 Regression Monte Carlo (Control randomization & Path recomputation)

-
1. At time $t_N = T$
Set $S(t_N, \tilde{X}_{t_N}^m) := g(t_N, \tilde{X}_{t_N}^m), m = 1, \dots, M$
 2. At time $t_i, i = N - 1, \dots, 1$
 - (a) [Conditional expectation approximation (state & control)]
Approximate $\Phi_{t_i}(\tilde{X}_{t_i}, \tilde{\alpha}_{t_i}) := \mathbb{E}[S(t_{i+1}, \tilde{X}_{t_{i+1}}) | \tilde{X}_{t_i}, \tilde{\alpha}_{t_i}]$ by regressing the sample $\{S(t_{i+1}, \tilde{X}_{t_{i+1}}^m)\}$ against $\{(\tilde{X}_{t_i}^m, \tilde{\alpha}_{t_i}^m)\}$ and call $\hat{\Phi}_{t_i}(\tilde{X}_{t_i}^m, \tilde{\alpha}_{t_i}^m)$ the resulting approximations.
 - (b) [Optimal strategy computation]
 $\tilde{X}_{t_i}^m \leftarrow \tilde{X}_{t_i}^m$ and $S(t_i, \tilde{X}_{t_i}^m) \leftarrow 0$.
For $t_j = t_i, \dots, t_N$:
 - i. Compute $\hat{\alpha}(t_j, \tilde{X}_{t_j}^m) := \arg \sup_{a \in A} \{f(t_j, \tilde{X}_{t_j}^m, a) \Delta t_j + \hat{\Phi}_{t_j}(\tilde{X}_{t_j}^m, a)\}$ using $\hat{\Phi}_{t_j}$.
 - ii. $S(t_i, \tilde{X}_{t_i}^m) \leftarrow S(t_i, \tilde{X}_{t_i}^m) + f(t_j, \tilde{X}_{t_j}^m, \hat{\alpha}(t_j, \tilde{X}_{t_j}^m)) \Delta t_j$.
 - iii. Update $\tilde{X}_{t_{j+1}}^m \leftarrow F(\tilde{X}_{t_j}^m, \hat{\alpha}(t_j, \tilde{X}_{t_j}^m), \Delta W_{t_j}^m)$.
 - (c) $S(t_i, \tilde{X}_{t_i}^m) \leftarrow S(t_i, \tilde{X}_{t_i}^m) + g(t_N, \tilde{X}_{t_N}^m).$
 3. At time $t_0 = 0$
 - (a) [Conditional expectation approximation (control only, as x_0 is deterministic)]
Approximate $\Phi_{t_0}(x_0, \tilde{\alpha}_{t_0}) := \mathbb{E}[S(t_1, \tilde{X}_{t_1}) | \tilde{\alpha}_{t_0}]$ by regressing $\{S(t_1, \tilde{X}_{t_1}^m)\}$ against $\{(\tilde{\alpha}_{t_0}^m)\}$.
 - (b) [Optimal strategy computation]
Compute $\hat{\alpha}(t_0, x_0) := \arg \sup_{a \in A} \{f(t_0, x_0, a) \Delta t_0 + \hat{\Phi}_{t_0}(x_0, a)\}$ and recompute \tilde{X} and S .
Then $\hat{v}(t_0, x_0) := \frac{1}{M} \sum_{m=1}^M S(t_0, \tilde{X}_{t_0}^m)$ is an estimate of $v(t_0, x_0)$
-

Let us summarize the method for the Euler-Maruyama scheme applied to equation (3.2). The application of the Euler scheme to this equation yields:

$$\tilde{x}_{t_{i+1}}^j = f(\tilde{x}_{t_i}^j, \varepsilon_i^j, \alpha_i^j) \quad (3.3)$$

$$f(x, \varepsilon, \alpha) := x + b(t_i, x, \alpha) \Delta t_i + \sigma(t_i, x, \alpha) \varepsilon \sqrt{\Delta t_i} \quad (3.4)$$

where t_i belongs to the time grid $\Pi = \{0 = t_0, t_1, \dots, t_N = T\}$ of the scheme, and $j \in [1, M]$ is the path index. For every i and j , ε_i^j is drawn from a Gaussian distribution, and α_i^j is drawn from the dummy distribution. Suppose that for any (ε, α) , the function $x \mapsto f(x, \varepsilon, \alpha)$ is invertible (call f_{inv} its inverse). Then, starting from the final value $x_{t_N}^j$ of the sequence (3.3), one can recover the whole trajectory of X :

$$x_{t_i}^j = f_{\text{inv}}(x_{t_{i+1}}^j, \varepsilon_i^j, \alpha_i^j) \quad (3.5)$$

provided one can recover the previous draws $(\varepsilon_{N-1}^j, \alpha_{N-1}^j), \dots, (\varepsilon_0^j, \alpha_0^j)$, which is easily achieved though the storage of the seeds of the random number generator, as shown by Algorithm 2 (initialized with $X[j] \leftarrow x_0 \forall j = 1, \dots, M$).

The first column of Algorithm 2 corresponds to the Euler scheme, with the addition of the storage of the seeds. At the end of the first column, the vector X contains the last values $X_T^j, j = 1, \dots, M$. From there, one can recover the previous values $X_{t_i}^j, i = N - 1, \dots, 0, j = 1, \dots, M$ as explained in the second column.

3.3 Adaptive local regression

The Regression Monte Carlo algorithm 1 requires a regression for each time step of the backward loop. How the regression is performed is therefore of utmost importance in practice. Consider the simple example of

Algorithm 2	Euler Scheme	Inverse Euler Scheme
1 % LOOP 1: Euler scheme		1 % LOOP 2: Inverse Euler scheme
2 for i from 0 to N-1		2 for i from N-1 down to 0
3 S[i] <- getseed()		3 setseed(S[i])
4 for j from 1 to M		4 for j from 1 to M
5 E <- randn(d)		5 E <- randn(d)
6 A <- randa(d)		6 A <- randa(d)
7 X[j] <- f(X[j], E, A)		7 X[j] <- finv(X[j], E, A)
8 end for		8 end for
9 end for		9 end for
10 S[N] <- getseed()		10 setseed(S[N])

valuing an American put option. Figure 3.1 displays the first regression of the backward loop (ie. performed on the last time step) for two different kinds of regression.

Figure 3.1a displays a standard regression with global polynomials, as proposed in Longstaff and Schwartz (2001). This method is simple and easy to implement, but has major drawbacks. Firstly, the boundaries cannot be fitted well, as the payoff is asymptotically linear, unlike polynomials of order higher than one. Secondly, and more problematically, the central part is not fitted well either, as a polynomial cannot reproduce an abrupt slope change very well. Using polynomials of higher degree does not mitigate these problems, unless the sample size is drastically increased at the same time.

Figure 3.1b displays a non-parametric locally linear regression, as defined for instance in Hastie et al. (2009), Chapter 6. The fit is spectacularly good, both the central part and the boundaries. Moreover, compared to other local regressions like the previously proposed piecewise regression on disjoint domains (Bouchard and Warin (2012), Jain and Oosterlee (2012)), there is no discontinuity problem. We argue that this kind of regression is the most suitable for the Regression Monte Carlo algorithm, especially when the payoff or intermediate cash flows are asymptotically linear (which is almost always the case in practice).

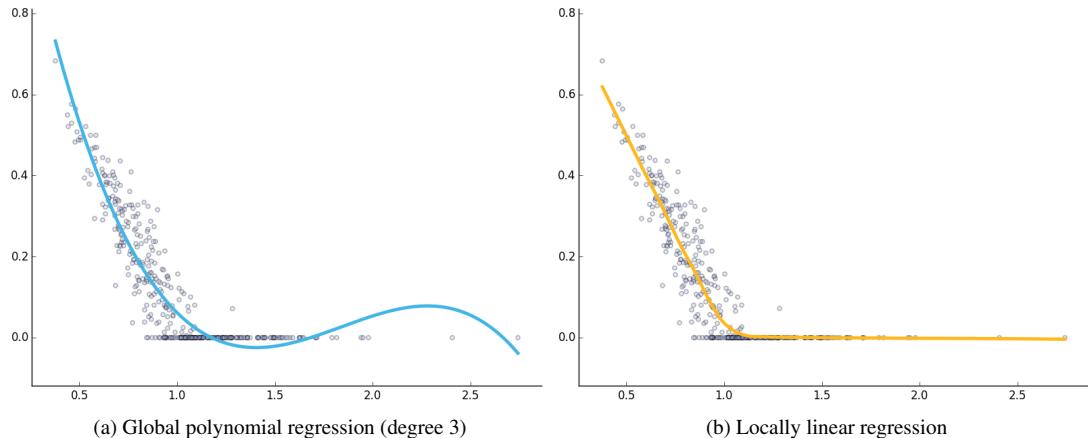


Figure 3.1. Global regression vs. local regression

The concern one may have with locally linear regression is about speed. Indeed, a naive implementation of it would require $\mathcal{O}(N^2)$ operations, where N is the number of points. Fortunately, there exists a smarter implementation to bring it down to a much faster $\mathcal{O}(N \log(N))$. We describe below the method for univariate data. The extension to multivariate data is developed in Langrené (2015).

Suppose that, given an input $\mathbf{x} = (x_1, \dots, x_N)^\top$, one wants to predict an output $\mathbf{y} = (y_1, \dots, y_N)^\top$ using (univariate) locally linear regression, ie. to compute, for each x_i , $1 \leq i \leq N$,

$$\min_{\alpha_i, \beta_i} \frac{1}{N} \sum_{j=1}^N K_h(x_i, x_j) [y_j - \alpha_i - \beta_i x_j]^2,$$

where K_h is a kernel function with bandwidth h . As shown for example in Chapter 6 of Hastie et al. (2009), computing the optimal regression coefficients $\hat{\alpha}_i$ and $\hat{\beta}_i$ boils down to compute, for every $1 \leq i \leq N$, sums of the type $\sum_{j=1}^N x_j^{p_x} y_j^{p_y} K_h(x_i, x_j)$ for integer powers $p_x \geq 0$ and $p_y \geq 0$. A naive implementation of these sums would require $\mathcal{O}(N^2)$ operations (for each x_i , compute $K_h(x_i, x_j)$ for every x_j). Fortunately, there exists a faster method to compute these sums. Suppose, without loss of generality¹, that the kernel function is the Epanechnikov kernel $K_h(x_i, x_j) = \frac{3}{4} \left[1 - \left(\frac{x_i - x_j}{h} \right)^2 \right] \mathbb{1}_{\{|x_i - x_j| \leq h\}}$. Then, developing $(x_i - x_j)^2$,

$$\sum_{j=1}^N x_j^{p_x} y_j^{p_y} K_h(x_i, x_j) = \frac{3}{4} \left(1 - \frac{x_i^2}{h^2} \right) S(x_i, p_x, p_y) + \frac{3}{2} \frac{x_i}{h^2} S(x_i, p_x + 1, p_y) - \frac{3}{4h^2} S(x_i, p_x + 2, p_y)$$

where $S(x_i, p_x, p_y) := \sum_{j=1}^N x_j^{p_x} y_j^{p_y} \mathbb{1}_{\{|x_i - x_j| \leq h\}}$. Now, suppose that the sample $(x_i)_{i=1, \dots, N}$ is sorted: $x_1 \leq x_2 \leq \dots \leq x_N$ (sorting costs $\mathcal{O}(N \log(N))$ operations). Then one can quickly compute $S(x_{i+1}, p_x, p_y)$ from $S(x_i, p_x, p_y)$, as

$$S(x_{i+1}, p_x, p_y) = S(x_i, p_x, p_y) - \sum_{j=1}^N x_j^{p_x} y_j^{p_y} \mathbb{1}_{\{x_i - h \leq x_j \leq x_{i+1} - h\}} + \sum_{j=1}^N x_j^{p_x} y_j^{p_y} \mathbb{1}_{\{x_i + h \leq x_j \leq x_{i+1} + h\}}.$$

This idea of “sliding the kernel” and updating the sums was proposed in Seifert et al. (1994) for univariate local regression, and extended to multivariate local regression in Langrené (2015), along with several practical improvements (scaling, change of basis, dimension reduction, adaptive bandwidth to nearest neighbours, shrinkage, etc.)

4 NUMERICAL RESULTS AND DISCUSSION

The implementation improvements described in Section 3.3 allowed us to successfully implement and solve the multi-dimensional stochastic control problem introduced in Section 2.

As many factors enter into play, involving many parameters, a thorough analysis of the outputs of the algorithm with sensitivity analyses is beyond the scope of this short paper, all the more so as the high-dimension makes it difficult, for instance, to display graphically the different regimes of the optimal policy, as was done for the two-dimensional mining problem studied in Chen et al. (2015). Instead, we will only provide a foretaste of the potential uses of the algorithm and, beyond valuation, the challenges ahead.

Table 4.1 provides the expected mean value and quantile values ($\mathbb{P}(V \leq q(x\%)) = x\%$) of the mining project described in Section 2 using the tools from Section 3, with the following parameters: $S_0 = 0.5$ (local currency), $\bar{S} = S_0$, $\kappa_S = 0.1$, $\sigma_S = \sqrt{0.08} \simeq 0.28$, $\lambda_s = 0.1$, $\sigma_{J_s} = 0.1$, $q_0 = 10$ (millions of tons), $\bar{q} = q_0$, $\kappa_q = 0.1$, $\sigma_q = 0.1$, $Q_0 = 150$ (millions of tons), $\sigma_Q = 0.1$, $\lambda_Q = 0.1$, $\sigma_{J_Q} = 0.1$

Table 4.1. Mean and quantiles of the mining project value (in millions)

q(1%)	q(10%)	q(25%)	q(50%)	mean	q(75%)	q(90%)	q(99%)
-5.9	-3.4	-1.8	0.9	8.1	11.4	27.7	80.9

The most striking observation from Table 4.1 is the positive skew of the mine value, as shown by the difference between the mean value (8.1) and median value (0.9) of the mine. This is not surprising, as similarly to Brennan and Schwartz (1985) and Chen et al. (2015) the possibility to abandon the mine limits the downside of the project value, but the inclusion of geological uncertainties reinforces this trend ($q(99\%) = 80.9 \simeq 10 \times \text{mean}$). This observation suggests that beyond a few risk factors, simple expectation of cash-flow as in (2.4) may be too rudimentary for project valuation. Risk preferences may be taken into account in (2.4) using existing methods (recursive utility, constrained optimization, viability, etc.). One could even include partial hedging against the uncertainties for which financial products are available (such as metal price).

Numerical methods from Section 3.3 make it possible to tackle high-dimensional stochastic control problem by regression Monte Carlo. Therefore, the remaining challenges do not lie in the computational side, but more in the modelling side. Beyond taking risk preferences into account, devising realistic dynamics for the risk

¹The methodology described in this subsection works for many kernel functions with finite support: uniform, triangular, biweight, triweight, etc.

factors may require more work, all the more so as the very long time horizon (several decades) suggests that some very long term macroeconomic effects should be factored in (evolution of world consumption, global depletion, etc.). Moreover, some risk factors such as estimated reserves would require data from mining companies, while such uncertainties as social license to operate remain a challenge to describe mathematically.

5 CONCLUSION

This paper describes several improvements to the Least-Squares Monte Carlo algorithm (control randomization, memory reduction, fast adaptive local regression) to make it suitable for high-dimensional stochastic control problems. We implemented this technique for a mine valuation problem, for which both price and geological uncertainties are considered. The method proposed can easily consider many more risk factors if needed. Our results suggest that realistic mine valuation under several sources of uncertainties is now computationally feasible, so that more effort can be spent on the modelling side of the problem. For mining companies, this suggests that real option algorithms are now ready to help them make the most out of their assets over time.

REFERENCES

- Aïd, R., L. Campi, N. Langrené, and H. Pham (2014). A probabilistic numerical method for optimal multiple switching problems in high dimension. *SIAM Journal on Financial Mathematics* 5(1), 191–231.
- Bao, C., M. Mortazavi, S. Northey, T. Tarnopolskaya, A. Monch, and Z. Zhu (2013). Valuing flexible operating strategies in nickel production under uncertainty. In *MODSIM2013, 20th International Congress on Modelling and Simulations*, pp. 1426–1432.
- Bouchard, B. and X. Warin (2012). Monte-Carlo valorisation of American options: facts and new algorithms to improve existing methods. In R. Carmona, P. Del Moral, P. Hu, and N. Oudjane (Eds.), *Numerical Methods in Finance*, Volume 12 of *Springer Proceedings in Mathematics*.
- Brennan, M. and E. Schwartz (1985). Evaluating natural resource investment. *Journal of Business* 58(2), 135–157.
- Castillo, F. and R. Dimitrakopoulos (2014). Joint effect of commodity price and geological uncertainty over the life of mine and ultimate pit limit. *Mining Technology* 123(4), 207–219.
- Chen, W., T. Tarnopolskaya, N. Langrené, and T. Lo (2015). Switching surfaces for optimal natural resource extraction under uncertainty. In *Proceedings of the 21st International Congress on Modelling and Simulation (MODSIM 2015)*.
- Dimitrakopoulos, R. and S. Abdel Sabour (2007). Evaluating mine plans under uncertainty: can the real options make a difference? *Resources Policy* 32, 116–125.
- Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer Series in Statistics. Springer.
- Hu, W. and S. Li (2014). The forward-path method for pricing multi-asset American-style options under general diffusion processes. *Journal of Computational and Applied Mathematics* 263, 25–31.
- Jain, S. and C. Oosterlee (2012). Pricing high-dimensional Bermudan options using the stochastic grid method. *International Journal of Computer Mathematics* 89(9), 1186–1211.
- Kharroubi, I., N. Langrené, and H. Pham (2014). A numerical algorithm for fully nonlinear HJB equations: an approach by control randomization. *Monte Carlo Methods and Applications* 20(2), 145–165.
- Langrené (2015). Fast multivariate local regression applied to local least-squares Monte Carlo.
- Longstaff, F. and E. Schwartz (2001). Valuing American options by simulation: a simple least-squares approach. *Review of Financial Studies* 14(1), 113–147.
- Ndiaye, A. and M. Armstrong (2013). Evaluating a small deposit next to an economically viable gold mine in west africa from the points of view of the mining company, the government and the local community. *Resources Policy* 38, 113–122.
- Seifert, B., M. Brockmann, J. Engel, and T. Gasser (1994). Fast algorithms for nonparametric curve estimation. *Journal of Computational and Graphical Statistics* 3(2), 192–213.