Extended Abstract: Utilizing Reinforcement Learning to Value a Multi-Factor Real Options Mining Project

Yuri Lawryshyn*and Reilly Pickard

Keywords: Real Options; Reinforcement Learning; American Options; Exercise Boundary Fitting; Project Valuation.

1 Introduction

Real option analysis (ROA) is recognized as a superior method to quantify the value of real-world investment opportunities where managerial flexibility can influence their worth, as compared to standard discounted cash-flow methods typically used in industry. The ability for managers to react to uncertainties at a future time adds value to projects, and since this value is not captured by standard DCF methods, erroneous decision making may result (Trigeorgis (1996)). A comprehensive ROA of an oil, gas or mineral mining project can improve the allocation of capital and managerial decision making and the methodology is currently used, to some degree, in the commodity extraction sectors. However, realistic models that try to account for a number of risk factors can be mathematically complex, and in situations where many future outcomes are possible, many layers of analysis may be required. Typically, managers are usually unable to understand the models and dismiss results that seem unintuitive to them.

An excellent empirical review of ex-post investment decisions made in copper mining showed that fewer than half of investment timing decisions were made at the right time and 36 of the 51 projects analyzed should have chosen an extraction capacity of 40% larger or smaller (Auger and Guzman (2010)). The authors were unaware of any mining firm basing all or part of their decision making on the systematic use of ROA and emphasize that the "failure to use ROA to assess investments runs against a basic assumption of neoclassical theory: under uncertainty, firms ought to maximize their expected profits". They make the case that irrational decision making exists within the industry due to a lack of real option tools available for better analysis. A number of surveys across industries have found that the use of ROA is in the range of 10-15% of companies, and the main reason for lack of adoption is model complexity (Hartmann and Hassan (2006), Block (2007), Truong, Partington, and Peat (2008), Bennouna, Meredith, and Marchant (2010), Dimitrakopoulos and Abdel Sabour (2007)).

Previously, we introduced a methodology based on exercise boundary fitting (EBF) in an effort to develop a practical Monte Carlo simulation-based real options approach (Bashiri, Davison, and Lawryshyn (2018)). We showed that our methodology converges in the case of simple Bermudan and American put options. More recently, we expanded on the model to solve a staged manufacturing problem (Fleten, Kozlova, and Lawryshyn (2021)). As we presented, utilizing boundary fitting

*Centre for Management of Technology and Entrepreneurship, Faculty of Engineering, University of Toronto, e-mail: yuri.lawryshyn@utoronto.ca

allowed us to solve a computationally difficult problem. In another study we explored the use of the EBF methodology for a number of cases, one being a build and abandon mining example (Davison and Lawryshyn (2021)). We showed that while the methodology provided good convergence on option value, under certain scenarios, where the optimal exercise boundaries occurred in regions where there were few Monte Carlo paths, the optimization algorithm struggled to converge. In Davison and Lawryshyn (2022) we explored convergence issues associated with the methodology and for the build and abandon mining example, we showed that by utilizing heuristic non-convex optimization, namely genetic algorithm (GA), we were able to circumvent the convergence issues, achieving satisfactory results. In Lawryshyn (2023) we presented preliminary results of utilizing reinforcement learning (RL) to solve the build / abandon problem. Our conclusion was that RL shows promise in solving such problems. In this study, we consider both one-factor and two-factor real options mining valuation problems. In the one-factor case, we further explore the use of RL to solve the build / abaondon problem we presented in Lawryshyn (2023). In the two-factor model we introduce a second process where the quantity (or quality) of the mined mineral is uncertain and, as mining proceeds, more is learned about the quantity available allowing for staged investment. We attempted to solve the two-factor "learn-as-you-go" (LAYG) problem using the EBF method previously (Davison and Lawryshyn (2019)) with partial success. Our objective is to explore the opportunity to use RL for valuing realistic, computationally difficult real options problems. We note that this study is a work in progress.

The rest of this paper is organized as follows. In the following section we provide a brief review of the literature. We first consider real options in the mining context, as we see this specific application an excellent test case for developing practical multi-factor real real option valuation methods, and then we present a brief review of the application of RL in option valuation. In Section 3 we present our methodology, first, framing the problem, then presenting our RL model. We present our preliminary results in Section 4, and discussion and conclusions in Section 5.

2 Relevant Literature

The academic literature is very rich in the field of mining valuation. Mining projects are laced with uncertainty and many discounted cash-flow (DCF) methods have been proposed in the literature to try to account for the uncertainty (Bastante, Taboada, Alejano, and Alonso (2008), Dimitrakopoulos (2011), Everett (2013), Ugwuegbu (2013)). Several guidelines/codes have been developed to standardize mining valuation (CIMVAL (2003), VALMIN (2015)). The main mining valuation approaches are income (i.e. cash-flows), market or cost based and the focus of this paper is on income-based real option valuation, which resemble American (or Bermudan) type financial options. Earlier real option works focused on modelling price uncertainty only (Brennan and Schwartz (1985), Dixit and Pindyck (1994), Schwartz (1997)), however the complexity in mining is significant and there are numerous risk factors. Simpler models based on lattice and finite difference methods (FDM) are difficult to implement in a multi-factor setting (Longstaff and Schwartz (2001)) and, also, it is extremely difficult to account for time dependent costs with multiple decision making points (Dimitrakopoulos and Abdel Sabour (2007)). Nevertheless, the simpler models continue to merit attention (Haque, Topal, and Lilford (2014), Haque, Topal, and Lilford (2016)). Dimitrakopoulos and Abdel Sabour (2007) utilize a multi-factor least squares Monte Carlo (LSMC) approach to account for price, foreign exchange and ore body uncertainty under multiple pre-defined operating

scenarios (states). However, the model only allows for operation and irreversible abandonment — aspects such as optimal build time, expansion and mothballing are not considered. Similarly, Mogi and Chen (2007) use ROA and the method developed by Barraquand and Martineau (2007) to account for multiple stochastic factors in a four-stage gas field project. Abdel Saboura and Poulin (2010) develop a multi-factor LSMC model for a single mine expansion. A review of 92 academic papers found that most real options research is focused on dealing with very specific situations where usually no more than two real options are considered (Savolainen (2016)). While the LSMC allows for a more realistic analysis, methods presented to date are applicable only for the case where changes from one state to another does not change the fundamental stochastic factors with time. For example, modular expansion would be difficult to implement in such a model if the cost to expand was a function of time and impacts extracted ore quantity due to the changing rate of extraction – these issues were considered in Davison, Lawryshyn, and Zhang (2015) and Kobari, Jaimungal, and Lawryshyn (2014). Also, modelling of multiple layers is still complex and will not lead to a methodology that managers can readily utilize.

Several recent papers apply RL to price/hedge financial options. In Pickard and Lawryshyn (2023) we presented a review of 17 recent papers related to the use of RL for hedging financial derivatives. An important result of the review was that RL trained agents, particularly those monitoring transaction costs, consistently outperform the Black-Scholes Delta method in frictional environments. The majority of the papers used actor-critic RL strategies to develop continuous hedging control. Recently, there have been a few studies using RL for real options analysis. For example, Lee, Chun, Roh, Heo, and Lee (2023) developed an RL framework of a real options problem valuation of a plant expansion in the context of carbon capture and utilization (CCU) where market demand and production uncertainty were considered. (Caputo and Cardin 2022) present an RL based approach to analyze flexibility in project valuation and evaluate a waste-to-energy system as a case study. These studies, as well as others, make the case for the use of RL to model complex real options that cannot be easily tackled with standard methods. The application of RL to price options is arguably in its infancy but shows significant promise and may prove to be an excellent solution methodology for complex RO with multiple factors.

Buehler, Teichmann, and Wood (2019) used RL and deep neural networks (NN) to approximate an optimal hedging strategy of a portfolio of derivatives considering market frictions. Their model outperformed simple delta hedging on a call option of the S&P500 index. Cannelli, Nuti, Sala, and Szehr (2022) formulated the optimal hedging problem as a risk-averse contextual k-armed bandit problem and showed that their model outperforms deep Q-networks (DQN) in terms of sample efficiency and hedging error when compared to delta hedging on simulated data. Cao, Chen, Hull, and Poulos (2021) used Q-learning and deep deterministic policy gradient (DDPG) to hedge a short position in a call option with transaction costs and showed reduced hedging costs compared to delta hedging on simulated data. Recent other studies applied several different RL models on simulated data and also showed superior performance to delta hedging (Kolm and Ritter (2019), Du, Jin, Kolm, Ritter, Wang, and Zhang (2020)). The application of RL to price options is arguably in its infancy but shows significant promise and may prove to be an excellent solution methodology for complex RO with multiple factors.

3 Methodology

In this section we introduce our methodology. In the following subsection we present the RL framework. Next, we frame the real option formulation for the one-factor build / abandon one-factor model and then build on this case to develop the two-factor model where we add the uncertainty associated with the quantity of the ore and allow for staged investment. Finally, we discuss the implementation of the Deep Q-Network (DQN) (Mnih, Kavukcuoglu, Silver, Graves, Antonoglou, Wierstra, and Riedmiller (2013)) to solve the real options problems.

RL Framework

RL is a sub-field of machine learning (ML) where the model automatically learns optimal decisions (actions) over time, typically in a stochastically changing environment. A comprehensive treatment of the topic is provided in Sutton and Barto (2018). As mentioned, this paper is a work in progress and our RL solution methodology will be based on Q-learning, specifically, the model-free framework based on DQN. Since our actions will be discrete for the problems considered here, the DQN framework is appropriate.

The RL solution framework is formulated within the Markov decision process (MDP) model (again, for details see Sutton and Barto (2018)). The MDP consists of an agent that interacts with its environment and is represented by states, actions and rewards. In the RL framework, the agent acts to optimize expected reward based on the current state of the environment. The agent develops a policy by learning to maximize the total expected reward, given current state and action pairs (Q-learning) by playing multiple games. In the context of real options, each game is represented by a Monte Carlo path of the underlying stochastic factor, or, in the case of a multi-factor setting, the (correlated) paths of the underlying stochastic factors. In the model-free setting, the model and optimal actions are learned through exploration of the environment. In the DQN framework, a (deep) neural network model is trained to provide actions to optimize the total expected reward for a given state.

One-Factor Build / Abandon Real Option

In the build / abandon real option valuation, we assume a stochastic process, X_t, that represents the price of the underlying mineral / material produced or mined, which we simulate using Monte Carlo simulation. We note that standard and non-standard processes can be used. There are four possible states related to the plant, namely,

- plant is not constructed,
- plant is under construction,
- plant is operating,
- plant has been abandoned.

Our RL state consists of X_t, $T-t$, where T is the terminal time for the valuation model, and two one-hot encoded variables, ξ_1 and ξ_2, where

$$\xi_1 = \begin{cases} 0, & \text{plant is not constructed} \\ 1, & \text{otherwise,} \end{cases}$$

$$\xi_2 = \begin{cases} 0, & \text{plant is operating} \\ 1, & \text{otherwise.} \end{cases}$$

Note that one and only one of $\xi_{i,t_j} = 1$ at any discretized time t_j. Furthermore, we do not need a state variable for the case where the plant is under construction nor abandoned. During the time the plant is being constructed we assume abandonment is not possible. Thus, as soon as the agent decides to construct, the episode time jumps ahead by the time to construct, τ_c, and enters the operating phase (state). This assumption ensures that the environment obeys the Markov property, thus ensuring the problem is a MDP, which RL solution methodologies are based on. We note that this assumption is not restrictive since the probability of the underlying process X_t reaching an abandon boundary in the time to construct after hitting the construction boundary is very low under normal circumstances. Thus, our state at time t_j consists of four variables, $S_{t_j} = \{X_{t_j}, T-t_j, \xi_{1,t_j}, \xi_{2,t_j}\}$.

Before construction, the agent must decide whether to do nothing, i.e. continue waiting (not exercise, action = 0) or construct (exercise, action =1). As discussed above, as soon as the decision to construct is made, the episode jumps to the operating state, so again, the agent must decide whether to continue operating (not exercise, action = 0) or abandon (exercise, action =1). Thus, the action space is $A_{t_j} = \{0, 1\}$.

The episode is terminated as follows:

- if $t_j >= T - \tau_c$ if the plant is not constructed,
- if action = 1 (abandonment) while the plant is operating ($\xi_{1,t_j} = 1$),
- if $t = T$ if the plant is operating.

At each t_j except during construction the reward is calculated as follows,

$$R_{t_j} = \begin{cases} -C_w \Delta t e^{-rt_j}, & \text{plant is not constructed and action = 0,} \\ -K e^{-rt_j}, & \text{plant is not constructed and action = 1,} \\ \gamma \left(X_{t_j}^{(i)} - C_{op} \right) \Delta t e^{-rt_j}, & \text{plant is operating and action = 0,} \\ -C_{op} e^{-rt_j}, & \text{plant is operating and action = 1,} \\ -C_{ab} e^{-rT}, & \text{plant is operating and } t_j = T, \end{cases} \tag{1}$$

where C_w is the cost rate of waiting (we use $C_w = 0$), K is the cost of construction, C_{op} is the operating cost rate, γ is the rate of production (extraction) of the product, Δt is the time step in the simulation and C_{ab} is the cost of abandonment.

Two-Factor Staged Investment Real Option

In the two-factor staged investment real option problem we utilize a stochastic price process, X_t, as above, but introduce a second standard Brownian motion, B_q, where q denotes the amount of

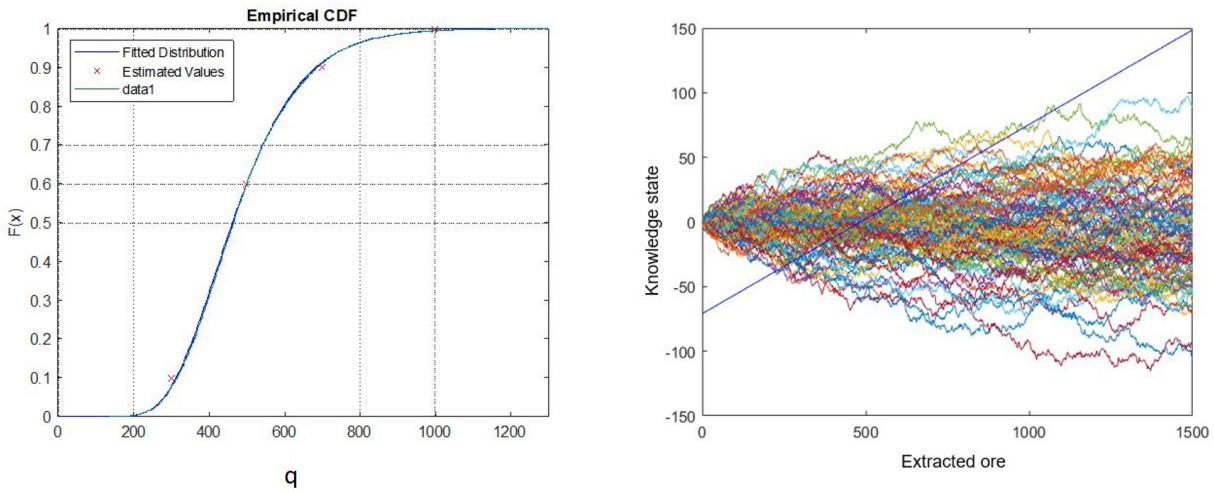

Figure 1: Example of managerial estimates and the best fit $F_{q^*}(q)$ distribution (right), and sample Brownian motion paths and the hitting line $-a - bq$.

material mined up to time, t. Our premise is that managers have an idea of how much ore is accessible when mining begins. As more ore is mined, managers gain a better understanding of ore quantity, thus "Learn as You Go", and can therefore expand or abandon operations optimally.

We assume that managers can provide estimates of ore quantity with certain levels of confidence, likely based on geological surveys of the site. For a standard Brownian motion, B_q, the distribution of the hitting *time* such that $\mathbb{P}(q^* \leq q)$ where $q^* = \min(q \geq 0,\ B_q = -a - bq)$ is

$$F_{q^*}(q) = e^{2ab}\Phi\left(\frac{-a - bq}{\sqrt{q}}\right) + 1 - \Phi\left(\frac{a - bq}{\sqrt{q}}\right), \tag{2}$$

where $\Phi(\cdot)$ is the standard normal distribution. We thus adjust the parameters a and b to match the distribution $F_{q^*}(q)$ as closely as we can to match the managerial estimates. An example is provided in Figure 3. The graph on the left plots managerial estimates of the amount of ore, represented by "x", and corresponding best fit $F_{q^*}(q)$ distribution. The graph on the right presents a sample of Brownian motion paths and the hitting line $-a - bq$.

We assume that the maximum plant capacity extraction rate is $\dot{Q}_{Max}$. The cost to build the skeleton of the plant is

$$K_0 = \alpha_M + \beta_M \dot{Q}_{Max} \tag{3}$$

where α_M and β_M are known fixed and variable cost components. We assume that the plant can be expanded by N modules such that each module would have $\frac{1}{N}\dot{Q}_{Max}$ capacity. The cost to n modules at any decision making time point is given as

$$K_n(n) = \alpha_n + \frac{n}{N}\beta_n \dot{Q}_{Max} \tag{4}$$

where α_n and β_n are other known fixed and variable cost components. Thus, to build a $n \leq N$ modules the first time, the cost would be $K_0 + K_n(n)$ and the subsequent addition of say $n' \leq N - n$

modules would be $K_n(n')$. Introducing the state variable $\xi_t \in 0, 1, 2, ..., N$ as the number of modules in operation, the extraction rate is

$$\dot{q}_t = \xi_t \dot{Q}_{Max} \tag{5}$$

and the total extracted quantity as

$$q_t = \int_0^t \xi_s \dot{Q}_{Max} ds. \tag{6}$$

Thus, for a given scenario, B_{q_t} defines our state of knowledge and if or when B_{q_t} hits the line $-a-bq_t$ line then the mine is depleted of ore. Furthermore, since B_{q_t} is observable, the RL agent will learn that when the B_{q_t} value is low, there is a greater chance that the mine will be depleted early, likely enhancing early exercise if the commodity price is low, and conversely when the B_{q_t} value is high, the agent will learn that early exercise may not be the best option.

Similar to the one-factor case, we assume that plant abandonment cannot take place while construction is on-going, and in this case, it would include both initial construction and expansion. We assume τ_{c_0} is the time for initial construction (time to build the plant skeleton and the first n modules) and τ_{c_n} is the time to construct extra n modules.

With a disrcitized time, t_j, the state consists of the following:

- X_{t_j}: Commodity price
- $T - t_j$: Time remaining in the analysis
- $B_{q_{t_j}}$: State of knowledge
- q_{t_j}: Amount of ore mined to date
- ξ_{t_j}: Number of modules in operation.

Similar to the one-factor case, before construction, the agent must decide whether to do nothing, i.e. continue waiting (not exercise, action = 0) or construct. However, at initial construction agent has a choice as to the number of modules to construct, ξ_{t_j}. After initial construction, which will require time τ_{c_0}, the agent can operate, construct more modules as long as $\xi_{t_j} < N$, idle or abandon the plant. We assume that expansion construction will not disrupt operations from the time a decision is made to add modules, during the period τ_{c_n}, however no further action is possible during this time. Thus, our discrete action space consists of a $N + 3$ actions, summarized as follows:

$$a_{t_j} = \begin{cases} \text{Wait to construct,} & \text{plant is not constructed,} \\ \text{Build initial } n \leq N \text{ modules,} & \text{plant is not constructed,} \\ \text{Operate plant,} & 1 \leq \xi_{t_j} \leq N, \\ \text{Expand plant by } n \leq N - \xi_{t_j} \text{ modules,} & 1 \leq \xi_{t_j} < N \text{ and plant is not under construction,} \\ \text{Idle the plant,} & 1 \leq \xi_{t_j} \leq N \text{ and plant is not under construction,} \\ \text{Abandon plant,} & 1 \leq \xi_{t_j} \leq N \text{ and plant is not under construction.} \end{cases}$$

We note that some actions listed above cannot be taken under certain states. For example, once we have expanded to ξ_{t_j} modules, we cannot expand by more than $N - \xi_{t_j}$ modules. In reinforcement learning, when dealing with discrete actions where some actions cannot be taken in certain states (referred to as invalid actions), there are a few common approaches to handle them.

- Masking: before selecting an action, one can apply a mask to remove invalid actions from consideration. This ensures that the agent only considers valid actions for each state.
- Penalizing: one can allow the agent to choose from all actions but penalize the reward heavily for choosing an invalid action. This encourages the agent to learn to avoid selecting invalid actions naturally.
- Dynamically adjusting the action space: one can dynamically adjust the action space based on the state. For example, if an action is invalid in a particular state, one can remove the invalid action from the action space for that state.
- Ignoring invalid actions: In some cases, it might be acceptable to simply ignore invalid actions and only consider valid actions. This approach works well if the number of invalid actions is small compared to the total number of actions.

In our case study, the total number of modules, N will be less than or equal to four therefore the method of ignoring invalid actions will be considered, as this method has been shown to work well when there are not many invalid actions. Masking has recently been studied and is an approach for us to consider in the future. Penalizing and dynamically adjusting the action space methods have generally shown inferior convergence.

At each t_j except during construction the reward is calculated as follows,

$$R_{t_j} = \begin{cases} -C_w \Delta t e^{-rt_j}, & \text{plant is not constructed,} \\ -(K_0 + K_n) e^{-rt_j}, & \text{plant is not constructed and action is to build initial } n \text{ modules,} \\ \xi_{t_j} \dot{Q}_{Max} \left(X_{t_j}^{(i)} - C_{op} \right) \Delta t e^{-rt_j}, & \text{plant is operating,} \\ -K_n(n) e^{-rt_j}, & \text{expand plant with } n \leq N - \xi_{t_j} \text{ modules,} \\ -C_i \Delta t e^{-rt_j}, & \text{plant is idling,} \\ -C_{ab} e^{-rt_j}, & \text{abandon plant,} \end{cases} \tag{7}$$

where C_w is the cost rate of waiting , C_{op} is the operating cost rate per unit produced, Δt is the time step in the simulation, C_i is the idling cost rate and C_{ab} is the cost of abandonment.

Deep Q-Network

In the DQN methodology, a neural network is used to to estimate the action value function, $Q(S, A)$. We used the keras-rl2 library. Our DQN agent was run with with the following parameters for the one-factor case:

- the model paremeter, i.e. the neural network, was setup as follows
  - number of inputs was set to one for the Bermudan put option, two for the American put option and four for the build / abandon real option (see above)
  - the neural network was fully connected with varying number of layers and the ReLU (rectified linear unit) activation function was used between layers, except for the last layer leading to the output nodes where a linear activation function was used

  - the output layer consisted of two outputs (as mentioned above, representing $Q(S, A)$ for the two possible actions, do not exercise or exercise)

- the memory parameter was set using the SequentialMemory option with a limit value of 50,000 and window size of 1
- the policy parameter was set to LinearAnnealedPolicy using the EpsGreedyQPolicy, with epsilon set to a maximum value of 1 and decaying to 0.1 over 5000 episodes
- the target_model_update was set to the default value of 0.01, so that the target network was updated at a rate of 1% of the total episodes taken during training.

As described above, each training episode is based on a single random path for X_t and the number of such paths was varied.

A full description for the two-factor case will be presented soon.

4 Results

To be presented.

5 Conclusions

References

Abdel Saboura, S. and R. Poulin (2010). Mine expansion decisions under uncertainty. *International Journal of Mining, Reclamation and Environment 24*(4), 340–349.

Auger, F. and J. Guzman (2010). How rational are investment decisions in the copper industry? *Resources Policy 35*, 292–300.

Barraquand, J. and D. Martineau (2007). Numerical valuation of high dimensional multivariate american securities. *JOURNAL OF FINANCIAL AND QUANTITATIVE ANALYSIS 30*(3), 383–405.

Bashiri, A., M. Davison, and Y. Lawryshyn (2018). Real option valuation using simulation and exercise boundary fitting - extended abstract. In *Real Options Conference*.

Bastante, F., J. Taboada, L. Alejano, and E. Alonso (2008). Optimization tools and simulation methods for designing and evaluating a mining operation. *Stochastic Environmental Research and Risk Assessment 22*, 727?735.

Bennouna, K., G. Meredith, and T. Marchant (2010). Improved capital budgeting decision making: evidence from canada. *Management Decision 48*(2), 225–247.

Block, S. (2007). Are "real options" actually used in the real world? *Engineering Economist 52*(3), 255–267.

Brennan, M. J. and S. Schwartz (1985). Evaluating natural resource investments. *Journal of Business 58*(2), 135–157.

Buehler, H., G. Teichmann, and B. Wood (2019). Deep hedging. *Quantitative Finance 19*(8), 1271–1291.

Cannelli, L., G. Nuti, M. Sala, and O. Szehr (2022). Hedging using reinforcement learning: Contextual k-armed bandit versus q-learning,. *arXiv*.

Cao, J., J. Chen, J. Hull, and Z. Poulos (2021). Deep hedging of derivatives using reinforcement learning. *The Journal of Financial Data Science 3*, 10–27.

Caputo, C. and M. Cardin (2022). Systems design using decision rules and deep reinforcement learning. *144*(2).

CIMVAL (2003). Standards and guidelines for valuation of mineral properties. Technical report, Canadian Institute of Mining, Metallurgy and Petroleum.

Davison, M. and Y. Lawryshyn (2019). Exercise boundary fitting in real option valuation of complex mining investments. In *Real Options Conference*.

Davison, M. and Y. Lawryshyn (2021). Real option valuation of a mining project using simulation and exercise boundary fitting. In *Canadian Operations Research Conference*.

Davison, M. and Y. Lawryshyn (2022). Convergence of 'exercise boundary fitting? least squares simulation approach. In *Real Options Conference*.

Davison, M., Y. Lawryshyn, and B. Zhang (2015). Optimizing modular expansions in an industrial setting using real options. In *19th Annual International Conference on Real Options*.

Dimitrakopoulos, R. (2011). Stochastic optimization for strategic mine planning: A decade of developments. *Journal of Mining Science 47*(2), 138–150.

Dimitrakopoulos, R. and S. Abdel Sabour (2007). Evaluating mine plans under uncertainty: Can the real options make a difference? *Resources Policy 32*, 116?125.

Dixit, A. and R. Pindyck (1994). *Investment under Uncertainty*. Princeton University Press.

Du, J., M. Jin, P. Kolm, G. Ritter, Y. Wang, and B. Zhang (2020). Deep reinforcement learning for option replication and hedging. *The Journal of Financial Data Science,*, 44?57.

Everett, J. (2013). Planning an iron ore mine: From exploration data to informed mining decisions. *Issues in Informing Science and Information Technology 10*, 145–162.

Fleten, S.-E., M. Kozlova, and Y. Lawryshyn (2021). Real option valuation of staged manufacturing - extended abstract. In *Real Options Conference*.

Haque, M. A., E. Topal, and E. Lilford (2014). A numerical study for a mining project using real options valuation under commodity price uncertainty. *Resources Policy 39*, 115?123.

Haque, M. A., E. Topal, and E. Lilford (2016). Estimation of mining project values through real option valuation using a combination of hedging strategy and a mean reversion commodity price. *Natural Resources Research 25*(4), 459–471.

Hartmann, M. and A. Hassan (2006). Application of real options analysis for pharmaceutical R&D project valuation?empirical results from a survey. *Research Policy 35*, 343–354.

Kobari, L., S. Jaimungal, and Y. A. Lawryshyn (2014). A real options model to evaluate the effect of environmental policies on the oil sands rate of expansion. *Energy Economics 45*, 155–165.

Kolm, P. and G. Ritter (2019). Dynamic replication and hedging: A reinforcement learning approach. *The Journal of Financial Data Science*.

Lawryshyn, Y. (2023). Extended abstract: Using reinforcement learning in applied real options modelling. In *Real Options Conference*.

Lee, J., W. Chun, K. Roh, S. Heo, and J. Lee (2023). Applying real options with reinforcement learning to assess commercial ccu deployment. *77*.

Longstaff, F. and E. Schwartz (2001). Valuaing american options by simulation: A simple least-squares approach. *The Review of Financial Studies 14*(1), 113–147.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller (2013). Playing atari with deep reinforcement learning.

Mogi, G. and F. Chen (2007). Valuing a multi-product mining project by compound rainbow option analysis. *International Journal of Mining, Reclamation and Environment 21*(1), 50?64.

Pickard, R. and Y. Lawryshyn (2023). Deep reinforcement learning for dynamic stock option hedging: A review. *11*(24), 4943–4963.

Savolainen, J. (2016). Real options in metal mining project valuation: Review of literature. *Resources Policy 50*, 49–65.

Schwartz, E. (1997). The stochastic behaviour of commodity prices: implications for valuation and hedging. *Journal of Finance 52*(3), 923?973.

Sutton, R. and A. Barto (2018). *Reinforcement Learning: An Introduction, Second Edition*. Cambridge, MA: MIT Press.

Trigeorgis, L. (1996). *Real Options: Managerial Flexibility and Strategy in Resource Allocation*. Cambridge, MA: The MIT Press.

Truong, G., G. Partington, and M. Peat (2008). Cost-of-capital estimation and capital-budgeting practice in australia. *Australian Journal of Management 33*(1), 95–122.

Ugwuegbu, C. (2013). Segilola gold mine valuation using monte carlo simulation approach. *Mineral Economics 26*, 39–46.

VALMIN (2015). The valmin code. Technical report, Australasian Institute of Mining and Metallurgy and the Australian Institute of Geoscientists.