An agent based framework for open pit mine planning

Hooman Askari-Nasab and Kwame Awuah-Offei[1]
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

Abstract

Long term production scheduling optimization has been a challenging issue for the mining industry because of the size and complexity of the problem. The current planning algorithms have limitations addressing the stochastic variables underlying the mine planning problem. In this paper an intelligent agent-based mine planning framework based on reinforcement learning is introduced. The long term mine planning is modeled as a dynamic decision network. The intelligent agent interacts with the block model by means of stochastic simulation and employs Q-learning algorithm to learn the sequence of push-backs that maximizes the net present value of the mining operation. The intelligent open pit simulator, IOPS, was implemented with an object oriented design in Java®. A comparative application case study was carried out to verify and validate the models. The proposed method was used in planning an iron ore deposit and the results were compared to the Milawa scheduler used in Whittle® software. The outcome of the study demonstrated that the intelligent agent framework provides a powerful basis for addressing real size open pit mine planning problems.

1. Introduction

The mining industry is faced with ever increasing complexities due to intense global competition, lower grade mineral deposits, price volatility, and geological uncertainty. More rigorous algorithms and enhanced numerical techniques are required to overcome the complexities currently facing the mining industry. The mine planning process defines the ore body depletion strategy over time. The planning of an open pit mine considers the temporal nature of the exploitation to determine the sequence of block extraction in order to maximize the generated income throughout the planning period. The optimal plan must determine the optimized ultimate pit limits and the mining schedule but such an objective results in a computationally intractable problem. Whittle (1989) outlined the complexity of the problem as: (i) the pit outline with the highest value cannot be determined until the block values are known; (ii) the block values are not known until the mining sequence is determined; and (iii) the mining sequence cannot be determined unless a pit outline is

[1] Assistant Professor, Department of Mining & Nuclear Engineering, University of Missouri-Rolla, USA

available. The optimal final pit limit algorithms conventionally neglect the time dimension of the problem and search for an ultimate contour that maximizes the total sum of the profits of all the blocks in the contour. The extraction sequence is then decided within the predetermined final pit limits. The optimized schedule cannot be attained without examining all possible combinations and permutations of the extraction sequence. Therefore, the scheduling algorithms must be able to deal with limitations of computing resources, time and space.

Open pit mine planning studies typically have focused on one of two objectives: (i) maximization of the discounted present value of cash flows (Tolwinski and Underwood, 1992; Elveli, 1995; Erarslan and Celebi, 2001; Halatchev, 2005; Dagdelen and Kawahata, 2007), or (ii) optimization of the plant feeding conditions (Youdi et al., 1992; Chanda and Dagdelen, 1995; Rubio, 2006; Yovanovic and Araujo, 2007). Current production scheduling methods are not just limited to, but can be divided into: heuristic methods; parametric analysis; operations research methods; and artificial intelligence techniques. The most common operations research methods include: mixed integer programming (MIP) (Gershon, 1983; Dagdelen, 1985; Ramazan and Dimitrakopoulos, 2004; Dagdelen and Kawahata, 2007), dynamic programming (Onur and Dowd, 1993) , goal programming (Chanda and Dagdelen, 1995; Esfandiari et al., 2004), and branch and bound techniques (Caccetta and Hill, 2003). Mixed integer programming mathematical optimization models have the capability to consider multiple ore processors and multiple elements during optimization. This flexibility of mathematical programming models result in production schedules generating significantly higher net present value than those generated by the other traditional methods. However, MIP formulations for optimization of production scheduling require too many binary variables, which makes the MIP models almost impossible to solve for actual open pit mining operations (Ramazan et al., 2005). Artificial intelligence methods such as machine learning expert system concepts (Tolwinski and Underwood, 1992; Elveli, 1995); genetic algorithms (Denby and Schofield, 1994; Denby et al., 1996; Wageningen et al., 2005); and applications of neural networks (Achireko and Frimpong, 1996; Frimpong and Achireko, 1997) have also been used to address the mine planning problem.

The key limitations of current mine planning methods are (i) inability to solve actual size mine problems; (ii) limitation in dealing with stochastic processes governing ore reserves, commodity price, cut-off grade, and production costs; (iii) inadequacy of the current final pit limits optimization techniques in taking into account the time aspect of exploitation; and (iv) shortcoming in defining the economics of ore with respect to the economics of the entire mining process, from ore to the finished product.

Research advances have led to concrete proposals and early applications of intelligent agents in mine planning and design (Askari-Nasab et al., 2005; Askari-Nasab and Szymanski 2007). The primary objective of this paper is to review the development of an intelligent agent-based theoretical framework for real size open pit mine planning. The study is a hybrid research work comprising algorithm development based on reinforcement learning concepts (Watkins, 1989; Sutton and Barto, 1998), and algorithm implementation in Java® programming language. A stochastic simulation model based on modified elliptical frustum (Askari-Nasab et al., 2004; Askari-Nasab et al., 2007) has been developed and used to model the geometry of the open pit layout expansion. The simulator returns the amount of ore, waste and the annual cash flow of the operation. The long term planning of the open pit mine is modeled as a dynamic decision network. The intelligent agent interacts with the open pit environment through simulation and employs Q-learning algorithm (Watkins, 1989) to maximize the net present value of the mining operation. The developed algorithms are implemented and applied to a real-world mining operation. The numerical applications of the developed models are compared with the results of common software used in industry to verify and validate the models. Finally, the potential application of the mine planning framework and significance of the research in mine planning is discussed.

2. Intelligent open pit planning theoretical framework

The reinforcement learning problem is formalized by the interaction of two basic entities: the agent and the environment. The agent is the learner and decision-maker. The agent's environment is comprised of everything that it cannot completely control. Thus, the environment defines the task that the agent is seeking to learn. A third entity, the simulation, mediates the interactions between the agent and the environment. The agent takes sensory input from the environment, and produces output actions that affect it. The interaction is usually an ongoing non-terminating process (Sutton and Barto, 1998).

Figure 1 illustrates the intelligent open pit optimal planning conceptual framework based on reinforcement learning terminology. The intelligent planning framework comprise independent, interactive and interrelated subsystems with processes, using reinforcement learning as the main engine to maximize the net present value of mining operations. The model illustrated in Figure 1 consists of three main entities of the reinforcement learning problem, agent, environment, and simulation. The main integral parts of the theoretical framework are as follows: (i) environment: consists of geological block model and economic block model; (ii) simulation: open pit production simulator that captures the discrete dynamics of open pit layout expansion, and materials transfer with the respective annual cash flows. The simulation model consists of a number of interrelated subsystems. The development and performance of the simulation components are discussed in (Askari-Nasab et al., 2004; Askari-Nasab, 2006; Askari-Nasab et al., 2007); (iii) agent: The simulated results are transferred to the intelligent open pit agent where Q-learning algorithm (Watkins, 1989) serves as the engine. The production simulator passes the respective amount of ore, waste, and the cash flows of the production periods to the agent. Development of the intelligent agent mine planning architecture is based on mathematically idealized forms of the reinforcement learning problem. The main concepts of optimality and the models in this study are developed and adapted from Sutton & Barto (1998) and Wooldridge (2002).

The reinforcement learning problem is meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The intelligent planning agent interacts with the block model through the production simulator and selects actions that are defined in terms of the changes in the push-back parameters and as the result, changes in the pit geometry. The simulation and the block model respond to those actions and present new possible pit push-backs to the agent. The open pit dynamics simulator in conjunction with the block model returns numerical rewards, which is the cash flow of each simulated production period. The primary goal of the agent is to maximize the NPV of the operation over time. This means maximizing not only the immediate reward, which is the cash flow of the next production period, but also the cumulative reward in the long run, which is the NPV.

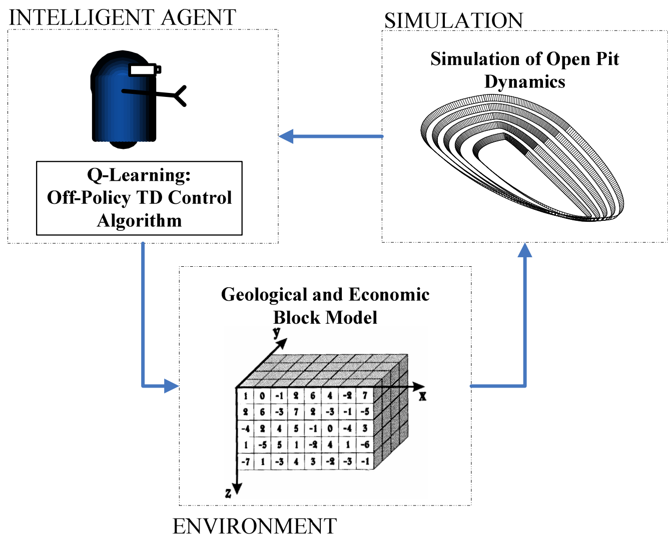


Figure 1- Intelligent open pit optimal planning frame work.

Figure 2 illustrates the mine planning intelligent agent architecture. The pit geometry evolution is viewed as series of snapshots over time. The agent and the simulation interact at each sequence of discrete time steps, $t = 1,...,n$. The simulation of the mining operation starts with the initial box cut at state, $s_{t-1} \in S_t$, and the agent responds by choosing the next pushback, $a_{t-1} \in A_t$, to be performed in this stage. Where S is the set of possible push backs, and $A(s_t)$ is the set of changes possible in the pit geometry in state s_t.

As a result of this action, the simulation and environment can respond with a number of possible states. However, only one state will actually result. On the basis of this second state of the environment, the agent again chooses an action to perform. The environment responds with one of a set of possible actions available, the agent then chooses another action, and so on. More specifically, the learning agent and simulation interact at each of a sequence of discrete time steps. At each time step t, the agent receives some representation of the open pit state, $s_t \in S$. On the basis of S, the agent selects an action, $a_t \in A(s_t)$, One time step later, in part as a consequence of its action, and interaction with the block model the agent receives a numerical reward, which is the cash flow of that period of mining operation, $r_{t+1} \in R$. As the result the agent finds itself in a new state, s_{t+1}. At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called the agent's policy and is denoted by, π_t, where $\pi_t(s,a)$ is the probability that $a_t = a$ if $s_t = s$.

Reinforcement learning methods specify how the agent changes its policy as a result of its experience. The agent's goal, roughly speaking, is to maximize the total amount of reward it receives over the long run. The objective is to maximize the *expected return*, where the return (see Figure 2), R_t given by Equation (1), is defined as a specific function of the immediate reward sequence. In Equation (1), γ is the discount factor and is a number between 0 and 1. The discount factor describes the preferences of an agent for current rewards over future rewards. When γ is close to 0, rewards in the distant future are viewed as insignificant. i in Equation (2) is the interest rate for time slice, t.

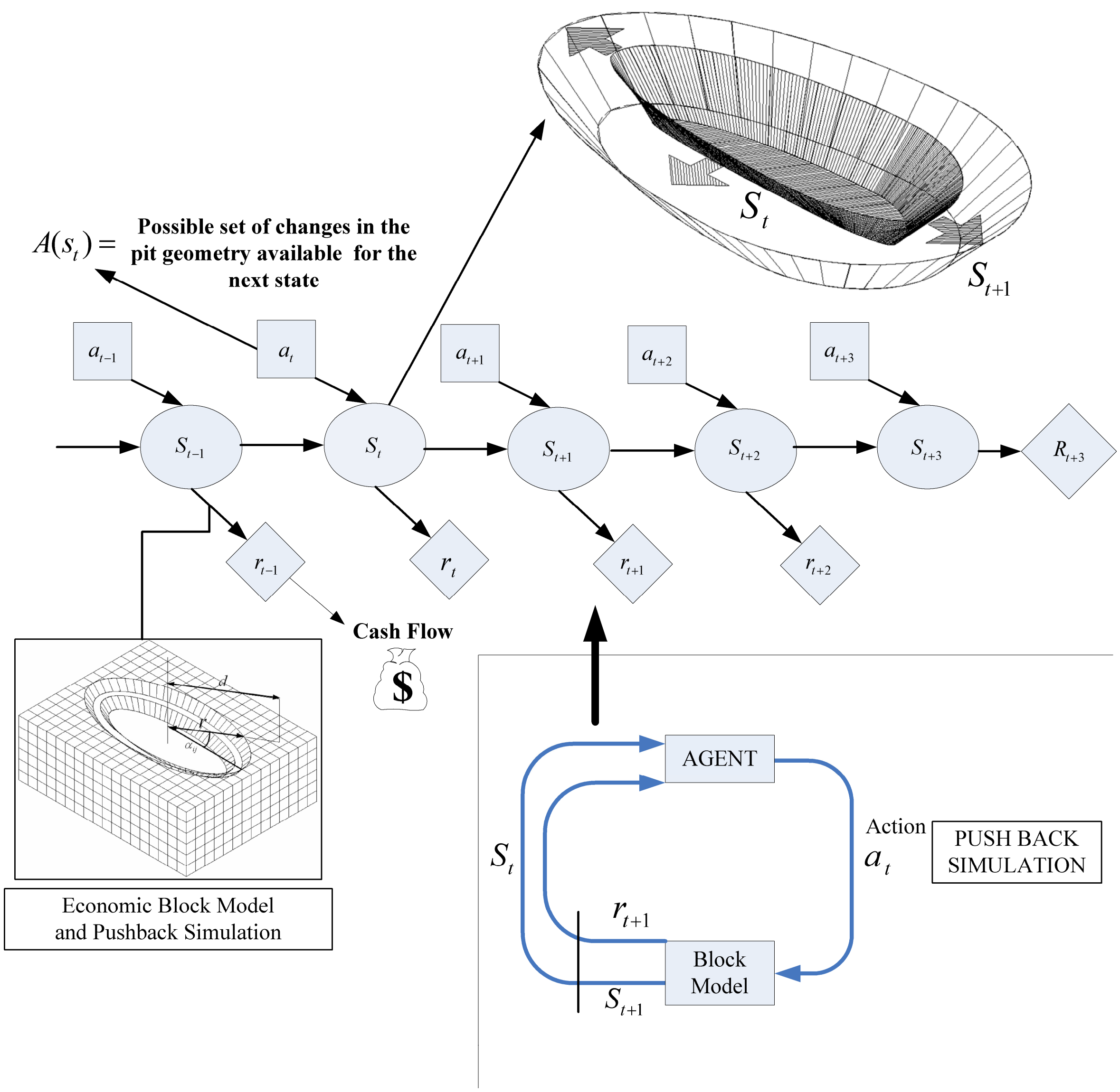


Figure 2 - Intelligent mine planning agent model as reinforcement learning problem.

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \gamma^k r_{t+k+1} = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \qquad (1)$$

$$\gamma = \frac{1}{1+i} \qquad (2)$$

Almost all reinforcement learning algorithms are based on estimating *value functions*--functions of states that estimate how good it is for the agent to be in a given state or how good it is to perform a given action in a given state. The notion of "how good" here is defined in terms of expected return. Accordingly, value functions are defined with respect to particular policies. Figure 3 illustrates a schematic of the open pit simulation at a discrete time step t and the open pit current status of S. For clarity of illustration it is assumed that there are just three possible push-backs a_1, a_2, a_3 that satisfy the targets of the next production period. Following one of the push-back designs the open pit will expand to the status of $s_1^{'}, s_2^{'}$, or $s_3^{'}$. The *value* of state s under policy π, denoted by $V^{\pi}(s)$, is the expected return or the NPV, when starting in s and following the policy thereafter, until reaching the final pit limits. For the Markov Decision Process representing the open pit dynamics in Figure 2, $V^{\pi}(s)$ can be defined as Equation (3).

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\} \quad (3)$$

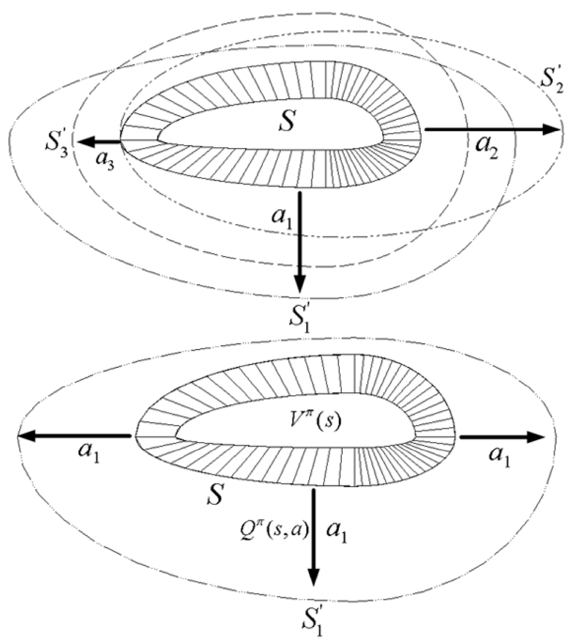


Figure 3 - Schematic of open pit simulation at a discrete time step t.

$E_{\pi}\{\ \}$ denotes the expected NPV given that the agent follows policy π, and t is any time step. The policy π is the current production schedule. The function V^{π} is called the *state-value function for policy* π. Similarly, the value of taking action a in state s under a policy π, denoted $Q^{\pi}(s,a)$ is defined as the expected NPV of the operation starting from s, taking the action a, and thereafter following the current schedule (policy π). Q^{π} is called the *action-value function for policy* π given by Equation (4).

$$Q^{\pi}(s,a) = E_{\pi}\{R_t \mid s_t = s, a_t = a\} = E_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\} \quad (4)$$

The Q-learning algorithm (Watkins, 1989) is used in this study to directly approximate Q^{π}, the optimal mine pushback design.

3. Algorithm development

Figure 4 illustrates the detailed flow chart of the intelligent optimal mine planning algorithm based on Q-learning algorithm (Watkins, 1989). The steps of the algorithm are as follows:

Step 1

The algorithm starts with (i) arbitrarily initializing the $Q(s,a)$, which is the expected discounted sum of future monetary returns of expanding the open pit from status S to the $S^{'}$ by choosing the push-back a and following an optimal policy thereafter; (ii) set the number of simulation trials that the algorithm is run. In other words the number of times that the open pit dynamics are being simulated from the initial box cut to the final pit limits.

Step 2

The push-back simulator captures the open pit layout evolution as a result of the material movement. At this stage the algorithm stochastically simulates a number of practical push-back designs for the next production period. The result of the simulation is k push-backs $a_1, a_2, ..., a_k$ that satisfy the tonnage production of the next period. Following each of these push-backs $a_1, a_2, ..., a_k$, the open pit will expand to the status of $s_1^{'}, s_2^{'}, ..., s_k^{'}$. The value of state s under policy π, denoted $V^{\pi}(s)$ is the expected return or the NPV of the sequence, when starting in s and following the policy thereafter until reaching the final pit limits.

Step 3

Simulated push-backs $a_1, a_2, ..., a_k$ are fitted on the economic block model, where the cash-flows $r_1, r_2, ..., r_k$ of each push-back are returned to the program.

Step 4

The epsilon greedy algorithm is called. The action selection rule is to select the action or one of the actions with highest estimated action value, that is, to select the push-back at time step t with the highest cash flow. The algorithm behaves greedily most of the time, which means it will select a push-back with the highest cash-flow among $r_1, r_2, ..., r_k$. But every once in a while, say with small probability ε, instead the algorithm selects an action at random, independently of the action-value estimates of the push-back. Subsequently the chosen push-back is implemented and the agent finds itself in pit status $S^{'}$ and observes the cash flow r.

Step 5

After being initialized to arbitrary numbers in step 1, Q-values $Q(s,a)$ are updated based upon previous experience as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a^{'}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (5)$$

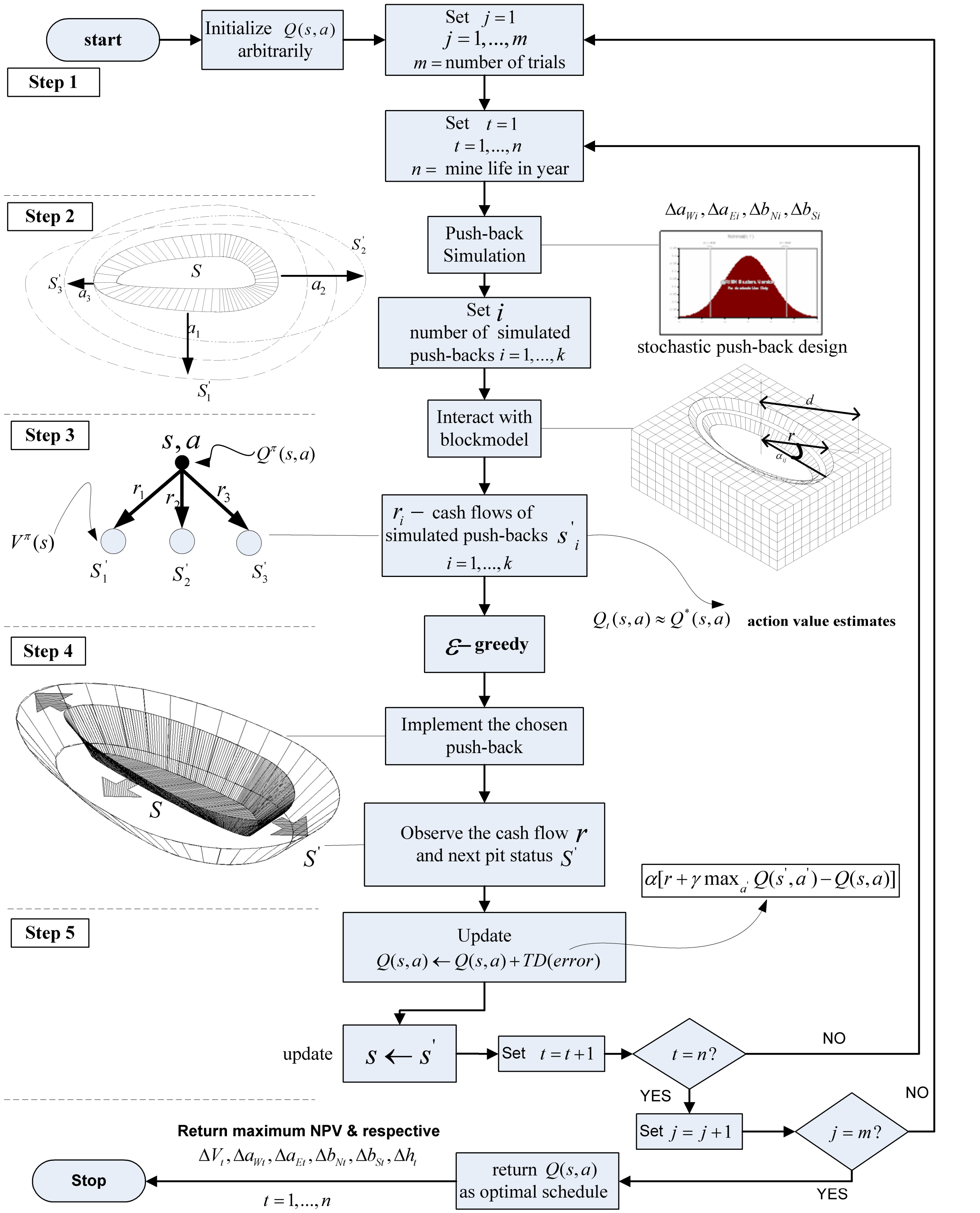
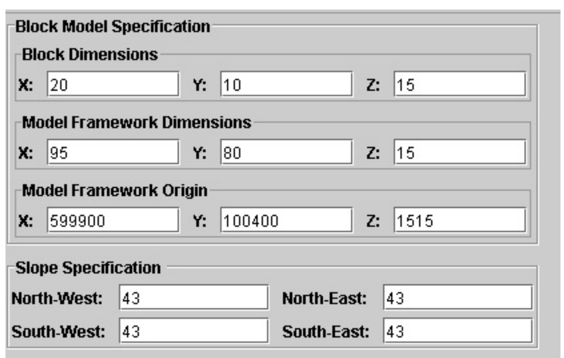


Figure 4 - Open pit Q-learning algorithm.

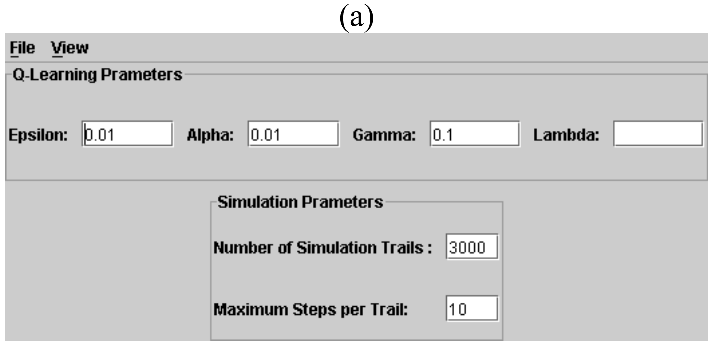
where: Q is the action-value function; α is a step-size parameter set to 0.01; S_t is the open pit geometrical state; a_t is a possible push-back at stage S; r_{t+1} is the cash flow of the simulated push-back; and γ is the discount factor. After updating the Q-values the algorithm moves to the next push-back and this process continues until it reaches the final pit limits. The algorithm will start the next episode of the push-back simulation by a random initial starting point in the pit. The number of iterations of simulation is controlled by the user. The algorithm is guaranteed to converge to the correct Q-values with the probability one under the assumption that the environment is stationary and depends on the current state and the action taken in it. Every state-action pair continues to be visited. Once these values have been learned, the optimal action from any state is the one with the highest Q-value.

4. Numerical Applications of the Intelligent Open Pit Simulator

A case study of an iron ore deposit is carried out to verify and validate the models. The extraction schedule from the Intelligent Open Pit Simulator is compared to the results of the Milawa algorithm and parametric analysis using Whittle® (Gemcom Software International, 1998-2006). The Intelligent Open Pit Simulator application was implemented in Java® (Sun Microsystems, 1994-2006) and MATLAB® (MathWorks, 2005) environment. This exercise consisted of class and object identification based on the Java Reinforcement Learning Library, JavaRL, (Kerr et al., 2003). The program requires the block model file as the input. The block model parameters are set through the block model specification tab illustrated in Figure 5(a). The Q-learning parameters and number of simulation iterations are set through the learning tab illustrated in Figure 5(b).



(a)



(b)

Figure 5 - (a) Block model specification (b) Q-learning parameters.

The iron ore deposit is explored with 159 exploration drill holes and 113 infill drill holes totalling 6,000 meters of drilling. Three types of ore, top magnetite; oxide; and bottom magnetite are classified in the deposit. Processing plant is based on magnetic separators so the main criterion to send material from mine to the concentrator is weight recovery. Kriging is used, to estimate the geological block model grades (Krige, 1951). The small blocks represent a volume of rock equal to 20 m×10 m×15 m. The model contains 114,000 blocks that makes a model framework with dimensions of 95×80×15. Figure 6 illustrates a multi cross-section of the deposit along sections 100100-east, 600245-north, and elevation of 1,590 m.

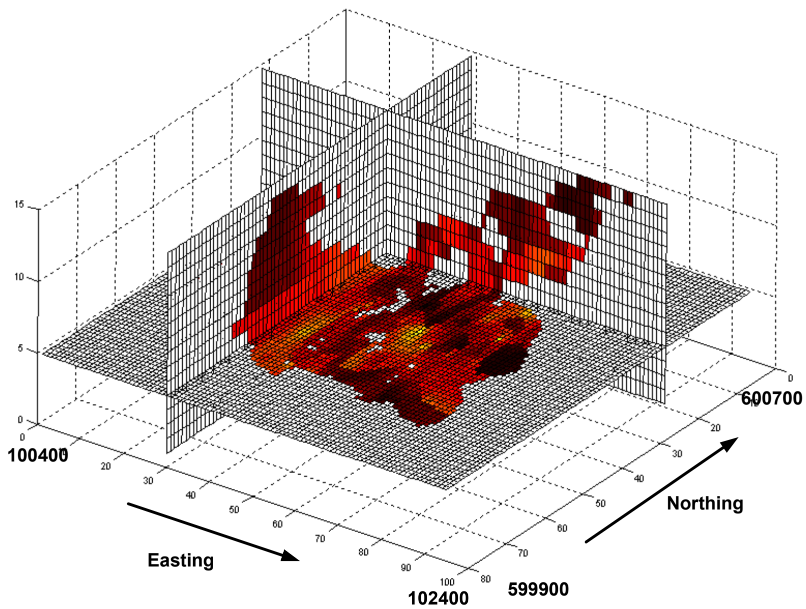


Figure 6- Three dimensional view of the deposit (coordinates in meters).

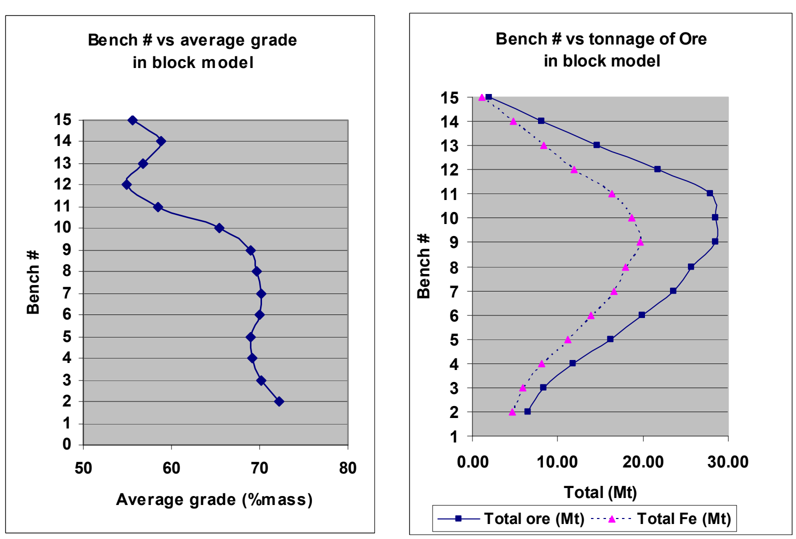


Figure 7 - Tonnage of ore and grade bench by bench.

The block model contains almost 243 million tonnes of indicated resource of iron ore with an average grade of 63%. Table 1 summarizes the block model information. Figure 7 shows the average grade, total amount of ore, and iron ore concentrate on a bench-by-bench basis.

The final pit limits are determined using the LG algorithm (Lerchs and Grossmann, 1965), using Whittle (Gemcom Software International, 1998-2006) software. Slope stability and geo-mechanical studies recommended a 43º overall slope in all regions. The average slope error in Whittle model is 0.9 degree and there are 35 possible structure arcs per block in the model which in total makes 3,075,666 arcs or edges in the graph model. The Pit Shells node in Whittle represents a set of pit shells generated by economic parametric analysis using the LG algorithm. This process reads in the block model from the Block Model node, pit slope constraints from Slope Set node, calculates block values using the economic and operational data contained in this node, and produces optimal pit outlines. The economic and mining parameters are based on: (i) mining cost = \$2/tonne; (ii) processing cost = \$2/tonne; (iii) selling price = \$15/tonne (Fe); (iv) maximum mining capacity = 20 Mt/year; (v) maximum milling capacity = 15 Mt/year; (vi) density of ore and waste = 4.2 tonne/m^3; and (vii) annual discount rate = 10%.

Table1- Summary of the ore and waste in the geological block model.

Rock Type	Blocks in model	Total (Mt)	Total Fe element (Mt)	Grade % Min	Grade % Avg	Grade % Max
Ore	19328	243.533	159.140	13	63.5	89
Waste	94672	1192.867	-	-	-	-

It is usual to produce multiple pit outlines in a single run and this process is controlled by the revenue factors in the optimization tab. The program finds a sequence of optimal push-backs based on varying the profitability of the deposit. In the generation of the pit shells, revenue factors in the range of 0.45 to 1.4 were used with variable geometric step sizes to scale base case price up and down, in order to control what nested pits are to be produced. It should also be considered that selection of a final pit has direct impact on the expected economic ore reserve. In terms of maximizing NPV, the lowest revenue factor that produces a pit sufficiently large to justify mining should also be the portion of the deposit to be mined first. Estimation of a project's NPV requires that timing of cash flow be accurately known so that an appropriate discount factor can be applied. This immediately introduces a problem for pit optimization software because the year of mining for any block of ore or waste will not be known until the mine production has been scheduled. The LG algorithm, which is the basis for Whittle software treats all mining activities as though it occurs simultaneously, with no discount factor applied. This usually results in selection of a final pit that is larger than the true maximum NPV pit.

Calculating the NPV requires knowing the relative time difference between blocks mined within a particular pit shell. This is dependent on the mill and mine capacities, practical sink rate (benches mined per year) and the equipment that can be practically operated within a specific cutback. Whittle provides a number of methods that work with the set of nested pits to provide a feasible production schedule. In this study the Milawa NPV algorithm was used. Milawa defines a variable bench interval between subsequent push-backs such that once a fixed number of benches have been mined out in the interior push-back then mining can commence on the next pushback. Thus, there is always a vertical lag of so many benches between push-backs. Milawa allows the lag to vary between push-backs and then searching for the combination of lags which is optimal either with respect to cash flow or managing stripping ratio.

The results of the Shells Node generated 77 nested pits with the respective total amount of ore, waste, and the NPV shown by Figure 8 for the best case, worst case, and Milawa algorithm. The appropriate push-backs are chosen in a way that the annual production targets are met in the long-term plan. The selected phases are represented by pits 17, 25, 43, 59, 65 and the final pit expected around pit 70. Successive schedules are run to different final pits from the first push-back to the pit shell number 77 in incremental steps of one. Pit shell number 68 with 209 million tonnes of ore and 182 million tonnes of waste has the highest NPV among all other pit shells and was chosen as the final pit limits for the production scheduling stage.

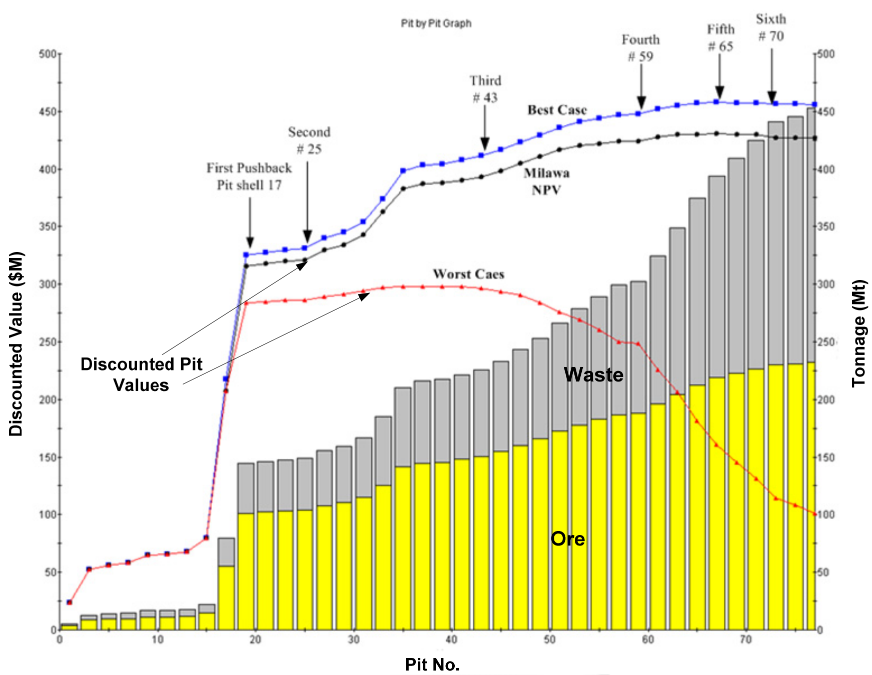


Figure 8 - Pit by pit graph.

The final pit outline in the previous section is the input for the comparison of Milawa NPV schedule and the Intelligent Agent algorithm. The comparative study is based on the following assumptions: (i) no stockpiles or materials re-handling was considered; (ii) blending of materials was not considered; (iii) the mill head grade and the annual mill feed was not set as a rigid constraint. The mill feed requirements are not the governing variables of the optimization in this case study; and (iv) all the planning parameters are kept the same in IOPS as the Whittle case study. The focus has been just on NPV maximization at this stage of the study. The final pit limits imported into IOPS are illustrated in Figure 9 with the respective dimensions of the major and minor axes of the frustum capturing the pit geometry. These dimensions are as follows: $a_W = 1,050\ m$; $a_E = 600\ m$; $b_N = 280\ m$; $b_S = 370\ m$; $h = 210\ m$.

The minimum mining width for the bottom of the pit was considered as an ellipse with major and minor axes of 60 m at any given time. The acceptable annual production targets were set to a maximum of 20 Mt; minimum of 19 Mt; and an average yearly production of 20Mt. IOPS simulates different mining starting points for each simulation episode based on a reference starting point coordinate provided by the user. Maximum three benches were allowed to be mined per year. The experiment was based on maximum mining capacity of 20 Mt/year and maximum milling capacity of 15 Mt/year. IOPS was used to run Q-learning algorithm with 3000 iterations with different scenarios of mining starting points. The probability that the agent "explores" as opposed to "exploiting" was set to $\varepsilon = 0.01$ in the epsilon-greedy algorithm. The learning rate for the intelligent agent, $\alpha = 0.01$; and the discount rate for delayed rewards, $\gamma = 0.1$.

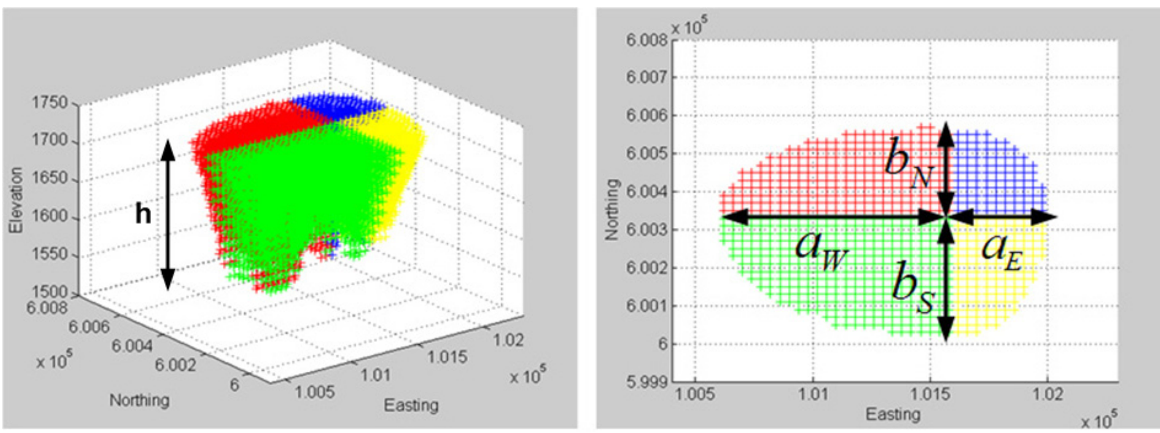


Figure 9 - Three-dimensional view and plan view of the final pit limits (meter).

5. Summary of results

The annual production schedule generated by IOPS compared to the results of Milawa NPV schedule are illustrated in Figure 10 and 11. From the analysis and comparisons of the results the following conclusions were drawn: (i) the optimized final pit limits show the total amount of 391 million tonnes of material consisting of 209 million tonnes of ore and 182 million tonnes of waste; (ii) Whittle 4-X yielded an NPV of $430 million over a 21-year of mine life at a discount rate of 10% per annum; (iii) IOPS yielded in an NPV of $438 million under the same circumstances and over the same mine life; (iv) The IOPS results proposed a starting point at 10160-east and 600340-north, which is located inside the smallest pit generated with nested pits in Whittle; (v) the fluctuations of annual production in both methods are caused by not setting the annual mill feed as the governing variable; (vi) IOPS shows a more consistent annual ore production compared to the Milawa NPV; and (vii) the Milawa NPV algorithm in Whittle 4-X is one of the standard tools widely used in industry.

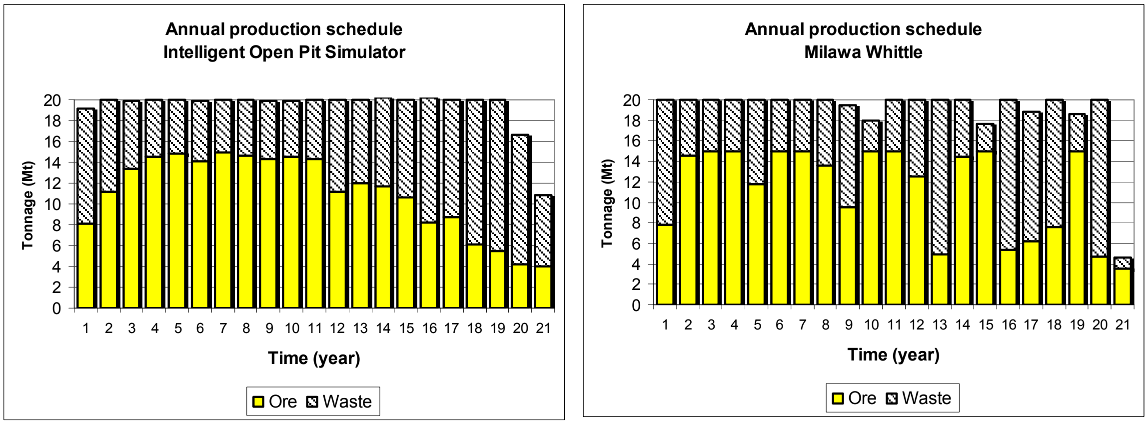


Figure 10- Comparative annual production schedule.

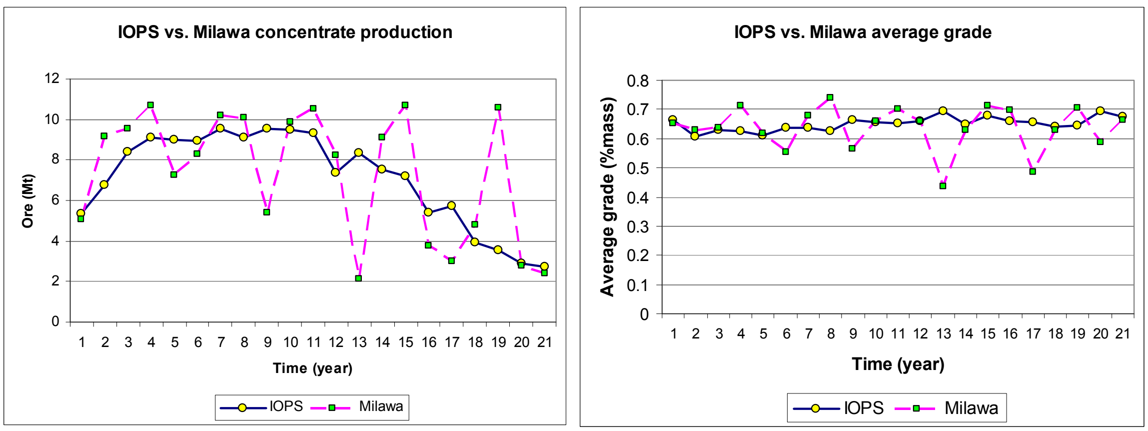


Figure 11- IOPS vs. Milawa results.

6. Conclusions

An intelligent agent theoretical framework for real size mine planning was developed based on reinforcement learning algorithms. The long term planning of the open pit mine is modelled as a dynamic decision network. The intelligent agent interacts with the open pit environment through simulation and employs Q-learning algorithm to maximize the net present value of the mining operation. An intelligent open pit production simulator, IOPS, is developed and implemented in Java® and MATLAB®. A stochastic simulation model captures the dynamics of open pit layout expansion. The developed algorithms are applied to a real-world mining operation. The numerical applications of the developed models are compared with the industry standard algorithms used in Whittle software.

The optimized final pit limits show the total amount of 391 million tonnes of material consisting of 209 million tonnes of ore and 182 million tonnes of waste. Whittle® software yielded an NPV of $430 million over a 21-year of mine life at a discount rate of 10% per annum. IOPS generated an NPV of $438 million under the same conditions. The focus of the case study at this stage has been on verifying and validating the models, which has been successful. The NPV from the IOPS schedule shows that the intelligent agent framework provides a powerful basis for addressing the real size open pit mine planning problem. Further focused research is required to develop and test the models based on intelligent agents to include more critical mine planning variables such as: variable optimized cut-off grades, constant annual mill feed, blending parameters, and stockpiles. Stochastic simulation as one of the major entities of the developed models has the strength to address the random field and dynamic processes involved in mine planning. The intelligent agent framework has the potential to be used for the optimal integration of mining and mineral processing systems, and development of a framework to quantify uncertainty relevant to mine planning and engineering design.

7. References

[1] Achireko, P. K. and Frimpong, S., (1996), "Open Pit Optimization using Neural Networks on Conditionally Simulated Blocks", *in Proceedings of 26th Applications of Computers and Operational Research in the Mineral Industry*, © SME, University Park, Pennsylvania, pp. 137-144.

[2] Askari-Nasab, H., (2006), "Intelligent 3D interactive open pit mine planning and optimization", PhD Thesis Thesis, © University of Alberta, Edmonton, Pages 167.

[3] Askari-Nasab, H., Awuah-Offei, K., and Frimpong, S., (2004), "Stochastic simulation of open pit pushbacks with a production simulator", *in Proceedings of CIM Mining Industry Conference and Exhibition*, © Edmonton, Alberta, Canada, pp. on CD-ROM.

[4] Askari-Nasab, H., Frimpong, S., and Awuah-Offei, K., (2005), "Intelligent optimal production scheduling estimator", *in Proceedings of 32nd Application of Computers and Operation Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, USA, pp. 279-285.

[5] Askari-Nasab, H., Frimpong, S., and Szymanksi, J., (2007), "Modeling Open Pit Dynamics using Discrete Simulation", *International Journal of Mining, Reclamation and Environment*, Vol. 21, 1, pp. 35-49.

[6] Askari-Nasab, H. and Szymanski , J., (2007), "Open Pit Production Scheduling Using Reinforcement Learning", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 321-326.

[7] Caccetta, L. and Hill, S. P., (2003), "An application of branch and cut to open pit mine scheduling", *Journal of Global Optimization*, Vol. 27, November, pp. 349-365.

[8] Chanda, E. K. and Dagdelen, K., (1995), "Optimal blending of mine production using goal programming and interactive graphics system", *International Journal of Surface Mining Reclamation and Environment*, Vol. 9, pp. 203-208.

[9] Dagdelen, K., (1985), "Optimum multi-period open pit mine production scheduling by Lagrangian parameterization", Ph.D. Thesis Thesis, © Colorado School of Mines, Golden, CO,

[10] Dagdelen, K. and Kawahata, K., (2007), "Oppurtunities in Multi-Mine Planning through Large Scale Mixed Integer Linear Programming Optimization", *in Proceedings of 33rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 337-342.

[11] Denby, B. and Schofield, D., (1994), "Open-pit design and scheduling by use of genetic algorithms", *Transactions of the IMM Section A*, Vol. 103, January - April 1994,, pp. A21-A26.

[12] Denby, B., Schofield, D., and Hunter, G., (1996), "Genetic algorithms for open pit scheduling - extension into 3-dimensions", *in Proceedings of 5th International Symposium on Mine Planning and Equipment Selection*, © A.A.Balkema/Rotterdam/Brookfield, Sao Paulo, Brazil, pp. 177-186.

[13] Elveli, B., (1995), " Open pit mine design and extraction sequencing by use OR and AI concepts", *International Journal of Surface Mining. Reclamation and Environment*, Vol. 9, pp. 149-153.

[14] Erarslan, K. and Celebi, N., (2001), "A simulative model for optimum open pit design", *The Canadian Mining and Metallurgical Bulletin*, Vol. 94, October, pp. 59-68.

[15] Esfandiari, B., Aryanezhad, M. B., and Abrishamifar, S. A., (2004), "Open pit optimization including mineral dressing criteria using 0–1 non-linear goal programming", *Mining Technology, Transactions of the Institutions of Mining and Metallurgy*, Vol. 113, January, pp. A3-A13.

[16] Frimpong, S. and Achireko, P. K., (1997), "The MCS/MFNN Algorithm for Open Pit Optimization", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 11, pp. 45-52.

[17] Gershon, M., (1983), "Mine scheduling optimization with mixed integer programming", *Mining Engineering*, Vol. 35, pp. 351-354.

[18] Halatchev, R. A., (2005), "A model of discounted profit variation of open pit production sequencing optimization", *in Proceedings of Application of Computers and Operations Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, pp. 315-323.

[19] International, G. S., (1998-2006), "Whittle strategic mine planning software", ver. 4.0: Gemcom Software International Inc.

[20] Kerr, A. J., Neller, T. W., Pilla, C. J. L., and Schompert, M. D., (2003), "Java Resources for Teaching Reinforcement Learning", *in Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '03)*, © Computer Science Research, Education, & Applications (CSREA) Press, Las Vegas, Nevada, pp. 1497-1501.

[21] Krige, D. G., (1951), "A statistical approach to some basic mine valuation and allied problems at the Witwatersrand",Thesis, © University of Witwatersrand, South Africa,

[22] Lerchs, H. and Grossmann, I. F., (1965), "Optimum design of open-pit mines", *The Canadian Mining and Metallurgical Bulletin, Transactions*, Vol. LXVIII, pp. 17-24.

[23] MathWorks, (2005), "MATLAB", ver. 7.04, MA, USA: MathWorks Inc.

[24] Onur, A. H. and Dowd, P. A., (1993), "Open pit optimization-part 2: production scheduling and inclusion of roadways", *Transactions of the Institution of Mining and Metallurgy*, Vol. 102, May-August, pp. A105-A113.

[25] Ramazan, S., Dagdelen, K., and Johnson, T. B., (2005), "Fundamental tree algorithm in optimising production scheduling for open pit mine design", *Mining Technology : IMM Transactions section A*, Vol. 114, 1, pp. 45-54.

[26] Ramazan, S. and Dimitrakopoulos, R., (2004), "Traditional and new MIP models for production scheduling with in-situ grade variability", *International Journal of Surface Mining, Reclamation & Environment*, Vol. 18, 2, pp. 85-98.

[27] Rubio, E., (2006), "Mill Feed Optimization for Multiple Processing Facilities using Integer Linear Programming", *in Proceedings of Proceeding of Fifteenth International Symposium of Mine Planning and Equipment Selection (MPES),*, © Torino, Italy, pp. 1207-1213.

[28] Sun Microsystems, I., (1994-2006), "Java Programming Language", ver. 1.4.2_08, 4150 Network Circle, Santa Clara, CA, USA

[29] Sutton, R. S. and Barto, A. G., (1998), "Reinforcement Learning, An Introduction", © The MIT Press, Cambridge, Massachusetts, Pages 432 pp.

[30] Tolwinski, B. and Underwood, R., (1992), "An algorithm to estimate the optimal evolution of an open pit mine", *in Proceedings of 23rd APCOM Symposium*, © SME, Littleton, Colorado, University of Arizona, pp. 399-409.

[31] Wageningen, A. V., Dunn, P. G., and Muldowney, D. M., (2005), "sequence optimization for long-term mine planning", *in Proceedings of 32 nd Application of Computers and Operation Research in the Mineral Industry*, © Taylor & Francis Group, London, Tucson, Arizona, USA, pp. 667-673.

[32] Watkins, C. J. C. H., (1989), "Learning from delayed rewards", PhD thesis Thesis, © University of Cambridge,

[33] Whittle, J., (1989), "The facts and fallacies of open-pit design," in *Manuscript, Whittle Programming Pty Ltd.* North Balwyn, Victoria, Australia

[34] Wooldridge, M., (2002), "An Introduction to Multi-Agent Systems", © John Wiley and Sons Limited, Chichester,UK, Pages 348.

[35] Youdi, Z., Qingziang, C., and Lixin, W., (1992), "Combined approach for surface mine short term planning optimization", *in Proceedings of 23rd APCOM Symposium*, © SME, Colorado, pp. 499-506.

[36] Yovanovic, A. P. and Araujo, A. C. d., (2007), "Operational Model and Computational Intelligence: A New Approach to Mineral Processing Optimization", *in Proceedings of 33 rd International Symposium on Computer Application in the Minerals Industry (APCOM)*, © GECAMIN LTDA, Santiago, Chile, pp. 491-498.