

Particle Swarm Procedure for the Capacitated Open Pit Mining Problem

Jacques A. Ferland¹, Jorge Amaya², Melody Suzy Djuimo¹

¹Département d'informatique et de recherche opérationnelle,

Université de Montréal, Montréal, Québec, Canada

ferland@iro.umontreal.ca, djuimoys@iro.umontreal.ca

²Centro de Modelamiento Matemático,

Universidad de Chile, Santiago, Chile

jamaya@dim.uchile.cl

Abstract

In the *Capacitated Open Pit Mining Problem*, we consider the sequential extraction of blocks in order to maximize the total discounted profit under an extraction capacity during each period of the horizon. We propose a formulation closely related to the *Resource-Constrained Project Scheduling Problem (RCPSP)* where the genotype representation of the solution is based on a priority value encoding. We use a *GRASP* procedure to generate an initial population (swarm) evolving according to a *Particle Swarm Procedure* to search the feasible domain of the representations. Numerical results are introduced to analyse the impact of the different parameters of the procedures.

Keywords: Open pit mining, GRASP, Particle Swarm, evolutionary process, RCPSP, priority encoding

1 Introduction

Consider the problem where a mining industry is analyzing the profit of extracting the ore contained in some site. In fact, the analysis includes two stages where two different problems have to be solved. The first problem is to determine the *maximal open pit* corresponding to the maximal gain that the mining industry can get from the extraction at the site. This problem can be formulated as identifying the *maximal closure* of an associated oriented graph. Picard proposes a very efficient procedure to solve this problem in [1].

The second problem, denoted *block extraction problem*, is to determine the extraction order leading to the maximum profit accounting for the discount factor and particular constraints related to the extracting operation. In general, the mining site is partitioned into blocks characterized by several numbers. One of these is the net value of the block estimated from prospect ion data. This net value is equal to the profit associated with the block corresponding to the difference between the ore content value and the cost of extracting the block. Hence this value can be negative if the ore content value is smaller than the cost of extracting it. Furthermore, the physical nature of the problem may require extracting blocks having negative net values in order to have access to valuable blocks.

In their pioneering work, Lerchs and Grossman [2] deals with this problem with an approach generating a sequence of nested pits. Then several other approach have been proposed : heuristics [3,4], Lagrangian

relaxation [3], parametric methods [5, 6, 7, 8], dynamic programming [9], mixed integer programming [3, 5, 10], and metaheuristics or artificial intelligence methods [11, 12]. In this paper, we rely on an analogy with the Resource-Constrained Project Scheduling Problem (RCPSP) [13] to propose a *particle swarm procedure* [14] to solve a specific variant of the problem referred to as the *Capacitated Open Pit Mining Problem*. Recall that a *particle swarm procedure* is an evolutionary population based process. The swarm (population) is evolving through the feasible domain searching for an optimal solution of the optimization problem. In the concluding remarks, we indicate how to extend it to other variants of this problem.

The paper is organized as follows. In Section 2, we formulate the two problems and we summarize Picard's procedure to solve the maximal open pit problem. Our solution approach relying on an analogy with the RCPSP to deal with the scheduling block problem is summarized in Section 3, and Section 4 includes our particle swarm procedure. In Section 5, we introduce numerical results to analyse the impact of the different parameters of the procedures. The numerical results allow verifying that the quality of the solutions generated with the particle swarm procedure is better than those obtained with greedy genotype priority vectors.

2 Problem formulations

In this section we first introduce a mathematical model for the maximal open pit problem and a method to solve it. Then the scheduling problem specifying the extraction order of the blocks is

formulated. The solution approach for this problem is given in the next section.

2.1 Maximal open pit

To formulate the maximal open pit model used to determine the maximal gain expected from the extraction, we denote by N the set of blocks in the mining site and by b_i the net value of extracting block $i \in N$. A binary variable x_i is associated with each block i :

$$x_i = \begin{cases} 1 & \text{if block } i \text{ is extracted} \\ 0 & \text{otherwise.} \end{cases}$$

The maximal open pit corresponds to the solution x maximizing the *objective function* $\sum_{i \in N} b_i x_i$.

The only physical constraints considered are related to the *maximal pit slope* leading to identify a set of predecessors B_i including the blocks of the preceding layer above i that have to be removed before block i can be extracted. The problem can be formulated using an oriented graph (*open pit graph*) $G=(V, A)$ where $V = \{i \in N : \text{node } i \text{ corresponds to block } i\}$ and the set of arcs $A = \{(i, j) : j \in B_i, i \in N\}$ used to indicate the precedence order of the block extraction. Hence the mathematical model associated with the maximal open pit problem can be summarized as follows

$$\begin{aligned} \text{(MOP)} \quad & \text{Max} \quad \sum_{i \in N} b_i x_i \\ \text{Subject to} \quad & x_j - x_i \geq 0 \quad (i, j) \in A \quad (1) \\ & x_i = 0 \text{ ou } 1 \quad i \in N. \quad (2) \end{aligned}$$

Now recall that the *closure* of an oriented graph [1] is a subset of nodes such that all successors of any of its node also belong to the set. It is easy to see that the problem (MOP) is equivalent to determine the *maximal closure* of the open pit graph $G=(V, A)$.

In [1], Picard shows that the problem of finding the *maximal closure* of $G=(V, A)$ is equivalent to determine a minimum cut (S, \bar{S}) of the associated Picard's graph $\bar{G}=(\bar{V}, \bar{A})$ specified as follows. The set on nodes $\bar{V} = V \cup \{s, t\}$ where s and t correspond to a super source and a super sink, respectively. The set of arcs $\bar{A} = A \cup A^+ \cup A^-$, where $A^+ = \{(s, i) : i \in V, b_i > 0\}$, $A^- = \{(i, t) : i \in V, b_i \leq 0\}$. Furthermore, the capacity d_{ij} of arc $(i, j) \in \bar{A}$ is specified as follows:

$$d_{ij} = \begin{cases} +\infty & \text{if } (i, j) \in A \\ b_j & \text{if } i = s \text{ and } (s, j) \in A^+ \\ -b_i & \text{if } j = t \text{ and } (i, t) \in A^- \end{cases}$$

Picard [1] shows that a maximal closure of G is the set $(S - \{s\})$. It follows that the Ford and Fulkerson labelling procedure [15] can be applied to determine the maximal flow in the Picard's graph associated with G , and then the maximal open pit is equal to $N^* = (S - \{s\})$ where (S, \bar{S}) is the minimal cut associated with the maximal flow.

2.2 Scheduling block extraction

The maximal open pit indicates the set of blocks to extract in order to maximize the total profit, but it does not give any indication of the order for extracting the blocs under operational constraints, nor does it account for the discount factor during the extraction horizon. This problem is more complex and hence more difficult to solve. In this paper, we consider the capacitated open pit mining problem where only one additional operational constraint related to the maximal quantity that can be extracted during each period of the horizon. In the concluding remarks we indicate how to extend the solution procedure to include other operational constraints.

Denote by p_i the weight of block $i \in N$, by C_t the maximal weight that can be extracted during period t , $1 \leq t \leq T$, and $\frac{1}{1+\alpha}$ the discount rate per period. A

binary variable x_i^t is associated with each block i for each period t :

$$x_i^t = \begin{cases} 1 & \text{if block } i \text{ is extracted during period } t \\ 0 & \text{otherwise.} \end{cases}$$

Referring to the notation introduce in Section 2.1, the scheduling block extraction problem can be formulated as follows:

$$\text{(SBE)} \quad \text{Max} \quad \sum_{t=1}^T \sum_{i \in N} \frac{b_i}{(1+\alpha)^{t-1}} x_i^t \quad (3)$$

$$\text{Subject to} \quad \sum_{t=1}^T x_i^t \leq 1 \quad i \in N \quad (4)$$

$$\sum_{j=1}^t x_j^t - x_i^t \geq 0 \quad (i, j) \in A, t = 1, \dots, T \quad (5)$$

$$\sum_{i \in N} p_i x_i^t \leq C_t \quad t = 1, \dots, T \quad (6)$$

$$x_i^t = 0 \text{ ou } 1 \quad i \in N, t = 1, \dots, T. \quad (7)$$

The objective function (3) accounts for the discount factor in evaluating the net values of the blocs when they are extracted. Constraints (4) guarantee that each block i is extracted at most once during the horizon. The extraction precedence is enforced by constraints (5), and the constraints (6) are related to the extraction capacity during each period of the horizon.

It can be shown that the set of blocks extracted in an optimal solution of (SBE) is a subset of the maximal

open pit. Hence, in the model (SBE), we can replace N by $N^* = (S - \{s\})$ determined in Section 2.1.

3 Solution approach

To find a schedule for the block extraction problem, we consider the similarity between (SBE) and the well known Resource Constraint Project Scheduling Problem (RCPS) [13]. The open pit mine extraction corresponds to the project, and the extraction of each block, to an activity of the project. The precedence relationship between the activities (blocks) is derived from the open pit graph $G = (V, A)$: for all $i \in N^*$

$$P_i = \{\text{block } i \text{ predecessors}\} = \{j \in N^* : (i, j) \in A\}.$$

The capacity constraints correspond to constraints (6) in problem (SBE). Hence, for any period t , let $E_t = \{i \in N^* : \text{block } i \text{ is extracted in period } t\}$, and the extraction capacity constraints can be written as

$$\sum_{i \in E_t} p_i \leq C_t \quad t = 1, \dots, T.$$

The reward associated with activity (block) i depends of the extraction period t : $\frac{b_i}{(1 + \alpha)^{t-1}}$.

The genotype representation $PR = [pr_1, \dots, pr_{|N|}]$ that we use is similar to the priority value encoding introduced by Hartman in [13] to deal with the RCPS. The i^{th} element $pr_i \in [0, 1]$ corresponds to the priority of scheduling block i to be extracted. Hence the priority of extracting block i increases with the value of pr_i , and these values are such that $\sum_{i=1}^{|N|} pr_i = 1$.

Starting from the genotype vector PR , a feasible phenotype solution x of (SBE) is generated using the following serial decoding scheme where the blocks are scheduled sequentially to be extracted. To initiate the first extraction period $t = 1$, we first remove the block among those having no predecessor (i.e., in the top layer) having the largest priority. During any period t , at any stage of the decoding scheme, the next block to be removed is one of those with the highest priority among those having all their predecessors already extracted such that the capacity C_t is not exceeded by its extraction. If no such block exists, then a new extraction period $(t + 1)$ is initiated. We denote by $v(PR)$ the value of the feasible solution x of (SBE) associated with the genotype vector PR . It is worth noticing that this decoding scheme is closely related to the *Serial SGS* procedure used to generate phenotype schedule from genotype representation for the RCPS.

In order to associate a priority to a block, we need to consider not only its net value b_i , but also its impact

on the extraction of other blocks in future periods. One such measure proposed by Tolwinski and Underwood in [9] is the *block lookahead value* \bar{b}_i . This value is determined by referring to the *spanning cone* SC_i of block i in the optimal open pit N^* specified as follows:

$$SC_i = \{j \in N^* : \text{there exists a path from } j \text{ to } i \text{ in } G\}.$$

$$\text{Hence, the value } \bar{b}_i = \sum_{j \in SC_i} b_j.$$

Referring to the *lookahead values*, several different genotype priority vectors can be generated using the following *GRASP procedure* [16] to determine sequentially the block priorities. At the k^{th} ($k = 1, \dots, N^*$) iteration, a block i_k is selected as follows:

- determine the subset of the $\beta\%$ best (in terms of their \bar{b}_i values) blocks not selected yet
- select randomly one block i_k in this subset.

$$\text{Then, } pr_{i_k} = \frac{(|N^*| - k + 1)}{1 + 2 + \dots + |N^*|} = \frac{2(|N^*| - k + 1)}{|N^*|(|N^*| + 1)}.$$

Note that the percentage $\beta\%$ is a parameter of the procedure.

Several feasible solutions of (SBE) can be generated by decoding different genotype vectors generated with the GRASP procedure. Furthermore, other process can be used to generate other genotype vectors. In the next section, we introduce an evolutionary process evolving in the set of genotype vectors in order to converge to an improved feasible phenotype solution of (SBE).

4 Particle Swarm Procedure

In this section, we introduce a particle swarm procedure [17] evolving through the set of genotype vectors in order to search the feasible domain of the problem (SBE). Consider a population (swarm) P of M genotype vectors (particles): $P = \{PR^1, \dots, PR^M\}$.

The M initial genotype vectors are generated using the GRASP procedure described in the preceding section. To initialize the first iteration of the particle swarm procedure, the genotype vector PR^k corresponds both to the current vector and the best achieved vector \overline{PR}^k of the individual k (i.e., at the start of the first iteration, $\overline{PR}^k = PR^k$). During the iterations of the procedure, the individuals (genotype vectors) of the population are evolving, and we denote by PRb the best overall genotype vector achieved so far. Hence, at the end of each iteration, $\overline{PR}^k, k = 1, \dots, M$, and PRb are updated as follows:

$$\overline{PR}^k := \text{ArgMax} \left\{ v(PR^k), v(\overline{PR}^k) \right\} \quad (8)$$

$$PRb := \text{ArgMax}_{1 \leq k \leq M} \left\{ v(\overline{PR}^k) \right\}. \quad (9)$$

Now we describe a typical iteration of the particle swarm procedure. Each current vector PR^k evolves individually to a new current genotype vector according to a probabilistic process accounting for its current value PR^k , for its best achieved value \overline{PR}^k , and for the best overall achieved vector PRb . More specifically, with each component i of each vector k , we associate a *velocity* [17] factor vc_i^k evolving at each iteration of the procedure. Its value is initialized at 0 (i.e., $vc_i^k = 0$) when the procedure starts. At each iteration, it evolves as follows:

$$vc_i^k := wvc_i^k + c_1 r_{1i}^k (\overline{pr}_i^k - pr_i^k) + c_2 r_{2i}^k (prb_i - pr_i^k), \quad (10)$$

and we define

$$ppr_i^k := vc_i^k + pr_i^k. \quad (11)$$

The probabilistic nature of the procedure follows from the fact that $r_{1i}^k \in [0, 1]$ and $r_{2i}^k \in [0, 1]$ are different uniform random number selected at each iteration. The impact of the terms is scaled by so called *acceleration coefficients* c_1 and c_2 . The *inertia weight* w was introduced in [18] to improve the convergence rate.

Using the vector $PPR^k = \left[ppr_1^k, \dots, ppr_{|N^*|}^k \right]$, we

determine the new current genotype PR^k . First, we have to verify if all components of PPR^k are non negative. Hence, determine

$$ppr^k = \text{Min}_{1 \leq i \leq |N^*|} \left\{ ppr_i^k \right\},$$

and if $ppr^k < 0$, then replace

$$ppr_i^k := ppr_i^k - ppr^k.$$

Thus the new current genotype vector PR^k is specified as follows:

$$pr_i^k := \frac{ppr_i^k}{\sum_{j=1}^{|N^*|} ppr_j^k}$$

where $pr_i^k \geq 0, i = 1, \dots, |N^*|$, and $\sum_{i=1}^{|N^*|} pr_i^k = 1$, as

required.

5 Numerical Results

Numerical results are now introduced to analyse the impact of the parameter values on the efficiency of the solution approach. The different procedures are implemented in Java 1.5, and the tests are executed on an AMD Athlon(tm) 64 Processor 3200, 2.2 GHz having 2.00 Go Ram.

We are using 20 different problems randomly generated over a two dimensions grid having 20 layers and being 60 blocks wide. For each problem, 10 clusters including the blocks having ore inside are randomly generated (note that these clusters can overlap). The value b_i of the blocks belonging to the clusters is selected randomly in the set $\{6, 8, 12, 16\}$. The rest of the blocks outside the clusters have no ore inside, and they have a negative value b_i equal to -4.

The optimal pit of each problem is determined with the approach described in Section 2.1. The number of blocks $|N^*|$ in the optimal pit of each problem is indicated in Table 1.

Table 1: Optimal pit sizes

Problem	$ N^* $	Problem	$ N^* $
1	64	11	291
2	78	12	307
3	88	13	320
4	90	14	353
5	95	15	382
6	113	16	385
7	156	17	428
8	158	18	434
9	205	19	495
10	230	20	508

The weight p_i of each block i is equal to 1, and the maximal weight $C_i = 3$ for each period.

We use the first 10 problems having smaller maximal open pit to analyse the impact of the different parameters. Each problem is solved using 12 different sets of parameter values as summarized in Table 2. Note that *iter* denotes the number of iterations of the particle swarm procedure.

For each set of parameter values l , each problem ρ is solved 5 times to determine

$va_{l\rho}$: the average of the best values $v(PRb)$ achieved

$vb_{l\rho}$: the best values $v(PRb)$ achieved

$\%o_{l\rho}$: the average % of improvement

$$\%o_{l\rho} = \frac{va_{l\rho} - v(PRb) \text{ at first iter.}}{v(PRb) \text{ at first iter.}} \times 100$$

$it_{l\rho}$: the last iteration where an improvement of PRb occurs.

Table 2: Parameter values

Set	β	M	Iter	w	c_1	c_2
1	10	50	20	0.7	1.4	1.4
2	50	50	20	0.7	1.4	1.4
3	100	50	20	0.7	1.4	1.4
4	10	25	20	0.7	1.4	1.4
5	50	25	20	0.7	1.4	1.4
6	100	25	20	0.7	1.4	1.4
7	50	50	20	0.1	1.4	1.4
8	50	50	20	0.7	0.7	1.4
9	50	50	20	0.7	1.4	0.7
10	50	25	20	0.1	1.4	1.4
11	50	25	20	0.7	0.7	1.4
12	50	25	20	0.7	1.4	0.7

Then for each set of parameter l , we compute the average values va_l , vb_l , $\%_l$, and it_l over the 10 problems. These values are summarized in Table 3, and they are used as criteria to evaluate the impact of the parameter values.

Table 3: Average values of criteria

Set	it_l	$\%_l$	va_l	vb_l
1	4.06	0.351	240.25	242.71
2	5.52	3.580	233.82	239.69
3	5.06	4.600	217.44	226.84
4	6.14	0.566	239.35	241.64
5	5.10	2.795	231.22	237.72
6	4.88	5.566	213.02	220.80
7	3.04	2.721	232.91	238.55
8	3.26	2.663	233.52	237.98
9	5.58	2.031	235.27	240.92
10	3.68	2.016	231.63	236.44
11	3.06	2.488	230.08	235.50
12	4.80	2.872	232.00	236.32

Impact of the $\beta\%$ in the GRASP procedure

Comparing rows 1, 2, and 3 of Table 3, we observe that the values of va_l and vb_l decrease while the value of $\%_l$ increases as the value of β increases. The same observations apply for rows 4, 5, and 6. Referring to the definition of the GRASP procedure in Section 3, it is easy to see that in general, the priority of blocks with larger lookahead values get larger when β decreases. Hence, we can expect the individuals in the initial population to have better profit values inducing that we can reach better solutions, but that the percentage of improvement $\%_l$ is smaller.

Impact of the size M of the population

The value va_l in row 1 is larger than in row 4. The same is true if we compare rows 2 and 5, and rows 3 and 6. This indicates that the values of the solutions generated are better when the size of the population is larger. This makes sense since we generate a larger number of different solutions.

Impact of the values of the particle swarm parameters

Comparing the results in rows 2, 7, 8, and 9, and those in rows 3, 10, 11, and 12, there is no clear impact of modifying the values of the parameters w , c_1 and c_2 .

Note that the values $w = 0.7$, $c_1 = 1.4$, and $c_2 = 1.4$ were selected accordingly to the authors in [19] who shown that setting the values of the parameters close to $w = 0.7298$ and $c_1 = c_2 = 1.49618$ gives acceptable results.

Note that the number of iterations it_l used by the particle swarm procedure to reach the best solution is smaller than the number $iter$ of iterations completed.

One could ask the following legitimate question. How the solutions obtained with an evolutionary process like particle swarm compare with the one obtained by decoding a greedy genotype priority vector generated with a variant of the GRASP procedure where, at each iteration, the block i with the largest value \bar{b}_i among those left is selected? To complete this analysis we use the last 10 problems (numbered from 11 to 20) having larger maximal open pit to obtain the results in Table 4. Using the parameter values in set 1, each problem ρ is solved 5 times to determine $va_{1\rho}$: the average of the best values $v(PRb)$ achieved for parameter values in set 1
 $vb_{1\rho}$: the best value $v(PRb)$ achieved for parameter values in set 1
 $vw_{1\rho}$: the worst value $v(PRb)$ achieved for parameter values in set 1
 v_{greedy} : the value obtained by decoding a greedy genotype priority vector.

This choice of the parameter values is justified by the results in Table 3 where the best value for va_l is obtained with the parameter values in set 1.

The results in Table 4 indicate that even the worst values $vw_{1\rho}$ is better than v_{greedy} for all problems. Furthermore, the percentage of improvement of $va_{1\rho}$ over v_{greedy} ranges from 2.32% to 52.24%. We can conclude that using particle swarm induces a gain in the solution quality.

Table 4: Improved solutions with particle swarm

Problem	$va_{1\rho}$	$vb_{1\rho}$	$vw_{1\rho}$	v_{greedy}
11	166.07	170.08	163.09	156.56
12	545.34	547.32	544.43	532.66
13	144.93	155.49	137.59	116.84
14	254.14	287.45	232.67	173.72
15	72.36	84.92	70.84	43.24
16	289.85	314.99	277.77	240.86
17	400.36	408.72	384.06	356.74
18	450.11	459.71	442.40	425.59
19	384.27	389.60	380.19	343.11
20	72.01	81.25	62.86	34.39

Referring to Table 4, we can also observe that for each problem, the interval $[vw_{1\rho}, vb_{1\rho}]$ including the values $v(PRb)$ is quite small. Indeed, the smallest ratio

$\Delta_\rho = \frac{vw_{1\rho}}{vb_{1\rho}}$ is equal to .774 for problem $\rho = 20$. This

indicates that the Particle Swarm procedure is stable.

6 Conclusion

In this paper we consider the Capacitated Open Pit Mining Problem where we have to determine the sequential extraction of blocks maximizing the total discounted profit under an extraction capacity during each period of the horizon. We propose a formulation closely related to the Resource-Constrained Project Scheduling Problem (RCPSP) where the genotype representation of the solution is based on a priority value encoding. We use a GRASP procedure to generate an initial population (swarm) evolving according to a Particle Swarm Procedure to search the feasible domain of the representations. The numerical results indicate that the better are the individuals in the initial population, the better is the solution generated but the smaller is the benefit of using an evolutionary process like particle swarm. Also, the quality of the solution increases with the size of the population. The numerical results in our experimentation do not seem to be very sensitive to the parameter values of the particle swarm procedure. Finally, the numerical results allow verifying that the quality of the solutions generated with the particle swarm procedure is better than those obtained with greedy genotype priority vectors.

Additional experimentations should be completed on problems of larger size closer to real world applications. Furthermore, it is easy to see that the decoding procedure of the genotype priority vectors can be adapted to account for additional constraints found in other variants of the scheduling block extraction problem.

References

- [1] Picard, J.C., "Maximal closure of a graph and applications to combinatorial problems", *Management Science*, 22, pp 1268-1272 (1976).
- [2] Lerchs, H., Grossman, I.F., "Optimum design for open pit mines", *CIM Bulletin*, 58, pp 47-54 (1965).
- [3] Cacetta, L., Kelsey, P., Giannini, L.M., "Open pit mine production scheduling", *Computer Applications in the Mineral Industries International Symposium (3rd Regional APCOM)*, Austral. Inst. Min. Metall. Publication Series, 5, pp 65-72 (1998).
- [4] Gershon, M., "Heuristic approaches for mine planning and production scheduling", *International Journal of Mining and Geological Engineering*, 5, pp1-13 (1987).
- [5] Dagdelen, K., Johnson, T.B., "Optimum open pit mine production scheduling by lagrangian parameterization", *Proceedings of the 19th APCOM Symposium of the Society of Mining Engineers (AIME)*, pp 127-142 (1986).
- [6] François-Bongarçon, D.M., Guibal, D., "Parametization of optimal designs of an open pit beginning a new phase of research", *Transactions SME, AIME*, 274, pp1801-1805 (1984).
- [7] Matheron, G., "Le paramétrage des contours optimaux", *Technical report no. 403*, Centre de Géostatistiques, Fontainebleau, France (1975).
- [8] Whittle, J., "Four-X user manual", *Whittle Programming Pty Ltd*, Melbourne, Australia (1998).
- [9] Tolwinski, B., Underwood, R., "A scheduling algorithm for open pit mines", *IMA Journal of Mathematics Applied in Business & Industry*, 7, pp 247-270 (1996).
- [10] Gershon, M., "Mine scheduling optimization with mixed integer programming", *Mining Engineering*, 35, pp 351-354 (1983).
- [11] Denby, B., Schofield, D., "The use of genetic algorithms in underground mine scheduling", *Proceedings 25th APCOM Symposium of the Society of Mining Engineers (AIME)*, pp 389-394 (1995).
- [12] Denby, B., Schofield, D., Bradford, S., "Neural network applications in mining engineering", *Department of Mineral Resources Engineering Mazine*, University of Nottingham, pp 13-23 (1991).
- [13] Hartmann, S., "A competitive genetic algorithm for the resource-constrained project scheduling", *Naval Research Logistics*, 456, pp 733-750 (1998).
- [14] Kennedy, J., Eberhart, R.C., "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks, IV*, pp 1942-1948 (1995).
- [15] Ford, L.R., Fulkerson, D.R., "Flows in Networks", Princeton University Press, New Jersey (1962).
- [16] Feo, T., Resende, M.G.C., "Greedy Randomized Adaptive Search Procedure", *Journal of Global Optimization*, 2, pp 1-27 (1995).
- [17] Paquet, U., Engelbrecht, A.P., "A new particle swarm optimiser for linearly constrained optimization", *Proceedings of the 2003 Congress on Evolutionary Computation*, pp 227-233 (2003).
- [18] Shi, Y., Eberhart, R.C., "A modified particle swarm optimizer", *Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, New Jersey*, pp 69-73 (1998).
- [19] Eberhart, R.C., Shi, Y., "Comparing inertia weights and construction factors in particle swarm optimization", *Proceedings of the Congress on Evolutionary Computation*, pp 84-88 (2000).