

Application of a Particle Swarm Algorithm to 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

²Departamento de Ingeniería Matemática and
Centro 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 [14].

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 [11] deals with this problem with an approach generating a sequence of nested pits. Then several other approach have been proposed: heuristics [1, 8], Lagrangian relaxation [1], parametric methods [2, 6, 12, 17], dynamic programming [16], mixed integer programming [1, 2, 7], and metaheuristics or artificial intelligence methods [3, 4].

In this paper, we rely on an analogy with the Resource-Constrained Project Scheduling Problem (RCPSP) [9] to propose a *particle swarm procedure* [10] 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.

2 Solution approach

The maximal open pit indicates the set of blocks to extract in order to maximize the total profit, accounting only for the constraints 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 . 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 our solution approach, we use the genotype representation $PR = [pr_1, \dots, pr_{|N|}]$ that is similar to the priority value encoding introduced by Hartman in [13] to deal with the RCPSP. 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 of the problem 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 associated with the genotype vector PR . 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 [16] is the *block lookahead value* \bar{b}_i . This value is determined by referring to the *spanning cone* SC_i of block i including all blocks requiring that block i be removed before they can be extracted.

A particle swarm procedure [13] is evolving through the set of genotype vectors in order to search the feasible domain of the problem. Consider a population (swarm) P of M genotype vectors (particles): $P = \{PR^1, \dots, PR^M\}$. The M initial genotype vectors are generated randomly using the GRASP to bias the values of the priorities in favour of the blocks having larger lookahead values. 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:

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

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

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* [13] 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:

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

and we define $ppr_i^k := vc_i^k + pr_i^k$.

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 [15] to improve the convergence rate.

Translating the vector $PPR^k = [ppr_1^k, \dots, ppr_{|N^*|}^k]$ to restore the non negativity of its component and normalizing the resulting vector, we determine the new current genotype PR^k .

3 Numerical Results

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 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 ranging from 64 to 230 blocks to analyse the impact of the different parameters. Each problem is solved using 12 different combinations of parameter values where $\beta\% \in \{10, 50, 100\}$ ($\beta\%$ being the parameter in the GRASP procedure to increase the bias of the priorities in favour of the blocks having larger lookahead values as its value decreases), $M \in \{25, 50\}$, $w \in \{0.1, 0.7\}$, and $c_1, c_2 \in \{0.7, 1.4\}$.

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

Then for each set of parameter l , we compute the average values va_l , vb_l , $\%_l$, and it_l over the 10 problems.

Impact of the size M of the population

The numerical results indicate 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

The numerical results indicate 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 [5] who shown that setting the values of the parameters close to $w = 0.7298$ and $c_1 = c_2 = 1.49618$ gives acceptable results.

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

From the numerical results, we observe that the values of va_l and vb_l decrease while the value of $\%_l$ increases as the value of $\beta\%$ increases. Since the priority of blocks with larger lookahead values get larger when $\beta\%$ decreases, 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.

To analyse the impact of increasing the number of individuals in the initial population that are generated with smaller values of $\beta\%$, we use the last 10 problems having larger maximal open pit ranging from 291 to 508 blocks. Furthermore, we set the parameter values to those generating the best results for the first 10 problems: $M = 50$, $w = 0.7$, $c_1 = c_2 = 1.4$. But for $\beta\%$ we consider the 6 scenarios summarized in Table 1 where

#10: the number of individual generated with $\beta\%=10$

#2 : the number of individual generated with $\beta\%=2$

#1 : the number of individual generated with $\beta\%=1$.

Each problem ρ is solved 5 times to determine:

$va_{sc\rho}$: the average of the best values $v(PRb)$ achieved for problem ρ using scenario sc

va_{sc} : the average value over the 10 problems.

Table 1: Scenarios for the initial population

Scenario	#10	#2	#1	va_{sc}
1	50	0	0	277.53
2	49	0	1	250.97
3	39	10	1	250.64
4	25	24	1	250.96
5	10	39	1	250.55
6	0	0	50	247.12

Referring to Table 1, it is worth noticing that the generation of the initial population becomes more greedy (the number of individuals in the initial population generated with smaller values of $\beta\%$ increases) as the scenario number increases. The results in Table 1 indicates that in general (except for scenario 4) the value of va_{sc} decreases as the scenario number increases. Furthermore, the increases is quite important when we move from scenario 1 to scenario 2 and from scenario 5 to scenario 6. Hence it seems appropriate to use smaller values of $\beta\%$ (like 10) to generate the initial population of individuals, but also to avoid being too greedy using very small values of $\beta\%$.

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 the GRASP procedure where $\beta\%=1$? The analysis is completed using the last 10 problems with the parameter values $M = 50$, $w = 0.7$, $c_1 = c_2 = 1.4$ and $\beta\% = 10$. Each problem ρ is solved 5 times to determine

$va_{1\rho}$: the average of the best values $v(PRb)$ achieved

$vb_{1\rho}$: the best value $v(PRb)$ achieved

$vw_{1\rho}$: the worst value $v(PRb)$ achieved for parameter values in set 1

v_{greedy} : the value obtained by decoding a greedy genotype priority vector.

The results in Table 2 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 1.03% to 56.56%. We can conclude that using particle swarm induces a gain in the solution quality.

Referring to this Table, 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 .809 for problem $\rho = 20$. This indicates

that the Particle Swarm procedure is stable.

Table 2: Improved solutions with particle swarm

Problem	va_{1p}	vb_{1p}	vw_{1p}	v_{greedy}
11	164.85	168.81	161.50	155.88
12	548.81	559.92	543.14	532.84
13	151.80	166.68	142.71	114.94
14	259.12	273.79	247.04	170.93
15	71.17	79.97	61.13	42.48
16	284.01	291.47	267.79	243.44
17	397.83	403.71	390.13	355.79
18	448.40	454.40	442.20	427.59
19	373.88	381.50	366.51	342.47
20	75.40	89.81	61.03	32.75

4 Conclusion

In this paper, we consider the Capacitated Open Pit Mining Problem to determine the sequential extraction of blocks maximizing the total discounted profit under an extraction capacity during each period of the horizon. We use a genotype representation of the solution based on a priority value encoding. An initial population (swarm) generated with a GRASP procedure is evolving according to a Particle Swarm Procedure. The numerical results indicate that the better are the individuals in the initial population, the better is the solution generated. Also, the quality of the solution increases with the size of the population. The numerical results do not seem to be very sensitive to the parameter values of the particle swarm procedure. Finally, the quality of the solutions generated with the particle swarm procedure is better than those obtained with greedy genotype priority vectors.

References

1. Cacetta, L., Kelsey, P., Giannini, L.M. (1998), "Open pit mine production scheduling", *Computer Applications in the Mineral Industries International Symposium (3rd Regional APCOM), Austral. Inst. Min. Metall. Publication Series*, 5: 65-72.
2. Dagdelen, K., Johnson, T.B. (1986), "Optimum open pit mine production scheduling by lagrangian parameterization", *Proceedings of the 19th APCOM Symposium of the Society of Mining Engineers (AIME)*, 127-142.
3. Denby, B., Schofield, D. (1995), "The use of genetic algorithms in underground mine scheduling", *Proceedings 25th APCOM Symposium of the Society of Mining Engineers (AIME)*, 389-394.

4. Denby, B., Schofield, D., Bradford, S. (1991), "Neural network applications in mining engineering", *Department of Mineral Resources Engineering Magazine*, University of Nottingham, 13-23.
5. Eberhart, R.C., Shi, Y. (2000), "Comparing inertia weights and construction factors in particle swarm optimization" *Proceedings of the Congress on Evolutionary Computation*, 84-88.
6. François-Bongarçon, D.M., Guibal, D. (1984), "Parametization of optimal designs of an open pit beginning a new phase of research", *Transactions SME, AIME*, 274: 1801-1805.
7. Gershon, M. (1983), "Mine scheduling optimization with mixed integer programming", *Mining Engineering*, 35: 351-354.
8. Gershon, M. (1987), "Heuristic approaches for mine planning and production scheduling", *International Journal of Mining and Geological Engineering*, 5: 1-13.
9. Hartmann, S. (1998), "A competitive genetic algorithm for the resource-constrained project scheduling", *Naval Research Logistics*, 45: 733-750.
10. Kennedy, J., Eberhart, R.C. (1995), "Particle swarm optimization", *Proceedings of the IEEE International Conference on Neural Networks, IV*: 1942-1948.
11. Lerchs, H., Grossman, I.F. (1965), "Optimum design for open pit mines", *CIM Bulletin*, 58: 47-54.
12. Matheron, G. (1975), "Le paramétrage des contours optimaux", *Technical report no. 403*, Centre de Géostatistiques, Fontainebleau, France.
13. Paquet, U., Engelbrecht, A.P. (2003), "A new particle swarm optimiser for linearly constrained optimization", *Proceedings of the 2003 Congress on Evolutionary Computation*, 227-233.
14. Picard, J.C. (1976), "Maximal closure of a graph and applications to combinatorial problems", *Management Science*, 22: 1268-1272.
15. Shi, Y., Eberhart, R.C. (1998), "A modified particle swarm optimizer", *Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, New Jersey*, 69-73.
16. Tolwinski, B., Underwood, R. (1996), "A scheduling algorithm for open pit mines", *IMA Journal of Mathematics Applied in Business & Industry*, 7: 247-270.
17. Whittle, J. (1998), "Four-X user manual" , *Whittle Programming Pty Ltd*, Melbourne, Australia.