

## Research Article

# Production Scheduling of Open Pit Mines Using Particle Swarm Optimization Algorithm

**Asif Khan and Christian Niemann-Delius**

*RWTH Aachen University, Institute of Surface Mining and Drilling (BBK III), Lochnerstraße 4-20, 52064 Aachen, Germany*

Correspondence should be addressed to Asif Khan; khan@bbk3.rwth-aachen.de

Received 3 July 2014; Revised 15 October 2014; Accepted 2 November 2014; Published 25 November 2014

Academic Editor: Imed Kacem

Copyright © 2014 A. Khan and C. Niemann-Delius. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Determining an optimum long term production schedule is an important part of the planning process of any open pit mine; however, the associated optimization problem is demanding and hard to deal with, as it involves large datasets and multiple hard and soft constraints which makes it a large combinatorial optimization problem. In this paper a procedure has been proposed to apply a relatively new and computationally less expensive metaheuristic technique known as particle swarm optimization (PSO) algorithm to this computationally challenging problem of the open pit mines. The performance of different variants of the PSO algorithm has been studied and the results are presented.

## 1. Introduction

Production scheduling of the open pit mines is a difficult and complex optimization problem. It aims to define the most profitable extraction sequence of the mineralized material from the ground that produces maximum possible discounted profit while satisfying a set of physical and operational constraints. Block model representation of the ore body commonly plays the role of the starting point for the planning and production scheduling of the open pit mines. The block model divides the mineralized body and the surrounded rock into a three-dimensional array of regular size blocks. A set of attributes, for example, grade, tonnage, density, and so forth, are then assigned to each one of these blocks estimated using some form spatial interpolation technique, for example, kriging, inverse distance weighting method, and so forth, and the exploratory drill hole sample data. The blocks are then divided into two categories, that is, ore and waste blocks. The blocks whose prospective profit exceeds their processing cost are categorized as an ore block to be sent for processing once mined while the rest are the waste blocks. An economic value is then assigned to each block by taking into account its group; that is, it is categorized as ore or waste, the commodity price, mining, processing, and marketing costs. The next step is to

define the final contour of the pit, that is, the limit to which it is economically feasible to mine by solving the ultimate pit limit problem (UPL). The mathematical formulation of the UPL problem can be defined as follows:

$$\begin{aligned} & \text{Maximize} \quad \sum_{i=1}^N v_i x_i \\ & \text{Subject to:} \quad x_i \leq x_j, \quad i = 1, 2, \dots, N; j \in P_i \\ & \quad \quad \quad x_i \in [0, 1], \end{aligned} \tag{1}$$

where  $v_i$  represents the economic value of block  $i$ ,  $N$  represents the total number of blocks in the block model,  $x_i$  represents a binary variable corresponding to block  $i$  which takes the value 1 if block  $i$  is inside the ultimate pit limits and 0 otherwise, and  $P_i$  represents the predecessor group of block  $i$ . The solution to this problem identifies the collection of blocks that will result into maximum possible undiscounted profit. Graph based Lerchs and Grossmann algorithm [1] or Max-flow algorithms [2] are commonly used for defining the ultimate pit limits. Depending on the size of the individual blocks and of the ore deposit, the UPL may contain thousands to millions of blocks that may have to be scheduled over

a time horizon typically ranging from 5 to 30 years while considering different physical and operational constraints which makes it a large combinatorial optimization problem.

This problem has been extensively discussed in the literature since 1960s and a wide variety of approaches have been proposed for solving it. Some of these techniques include parametric or ultimate pit limit (UPL) based approaches [3–5], integer programming [6, 7], dynamic programming [8, 9], and blocks aggregation/clustering [10]. The main limitations of these techniques are that either they have high computational cost when applied to real sized problem or they can only handle a simplified version of the original problem. In the recent years a new class of computationally less expensive metaheuristic techniques have attracted the attention of several researchers to solve this problem such as Genetic Algorithms [11–13], Simulated Annealing [14, 15], and Ant Colony Optimization [16, 17]. Though these techniques do not guarantee the optimality of the final solution that they produce, their lower computational cost makes them attractive alternative choice over the computationally expensive exact optimization algorithms. The objective of this paper is to propose a procedure to apply a similar metaheuristic technique known as particle swarm optimization (PSO) to the open pit mine production scheduling problem and to check and compare the efficiency of its different variants. In the past Ferland et al. [18] have proposed an approach to apply particle swarm optimization to the capacitated open pit mine production scheduling problem considering only one mining capacity constraint during each period of the mine life. They used a genotype representation of the solutions based on the priority value encoding and proposed a decoding scheme to generate feasible phenotype solutions from the genotype solutions. A GRASP procedure was suggested to produce the initial population of genotype solutions that evolve through the solution space using global version of particle swarm algorithm. Different properties of the procedure were studied by its application to a group of two-dimensional (2D) small problems. In this paper a totally different procedure has been proposed to apply the particle swarm optimization algorithm to the more general three-dimensional (3D) case of the open pit mine production scheduling problem with upper and lower bounds on the processing and mining capacity constraints in each period of the scheduling horizon.

The remainder of the paper is organized as follows. In Section 2 the general integer programming formulation of the production scheduling problem of the open pit mines is presented. In Section 3 the standard PSO algorithm is discussed. In Section 4 the proposed procedure for applying PSO algorithm to open pit production scheduling problem is presented in detail. The results of the numerical experiments are presented in Section 5 followed by the conclusions in Section 6.

## 2. Problem Formulation

The general integer programming formulation of the open pit production scheduling problem with the objective to maximize the net present value of the operation while satisfying a set of physical and operational constraints similar

to the one defined by Gaupp in [19] can be described as follows.

*Objective Function.* Maximize the net present value of the extraction sequence:

$$\text{Maximize} \sum_{i=1}^N \sum_{t=1}^T v_{it} x_{it}, \quad (2)$$

where

$v_{it} = v_i / (1 + d)^t$  is the discounted economic value of block  $i$  if mined in period  $t$ ;  $d$  represents the annual discount rate;

$x_{it} = \{1, \text{if block } i \text{ is mined in period } t; 0, \text{otherwise}\};$

$N$  = total number of blocks;

$T$  = total number of periods;

$t$  = time period index,  $t = 1, 2 \dots T$ ;

$i$  = block index,  $i = 1, 2 \dots N$ .

This is subject to the following.

*Reserve Constraints.* A block cannot be mined more than once:

$$\sum_{t=1}^T x_{it} \leq 1 \quad \forall i = 1, 2 \dots N. \quad (3)$$

*Slope Constraints.* Each block can only be mined if its predecessors are already mined in or before period  $t$ :

$$x_{it} - \sum_{\tau=1}^t x_{j\tau} \leq 0 \quad \forall i = 1, 2 \dots N; \quad t = 1, 2 \dots T; \quad (4)$$

where  $j \in (\text{set of predecessors blocks of block } i)$ .

*Mining Capacity Constraints.* The total material mined during each period should be within the predefined upper and lower mining limits:

$$\begin{aligned} \sum_{i=1}^N w_i * x_{it} &\geq \underline{M}^t, \quad \forall t = 1, 2 \dots T, \\ \sum_{i=1}^N w_i * x_{it} &\leq \overline{M}^t, \quad \forall t = 1, 2 \dots T. \end{aligned} \quad (5)$$

$\underline{M}^t, \overline{M}^t$  represent the lower and upper limits of the available mining capacity in period  $t$ , respectively (*constant*), and  $w_i$  represents the block tonnage.

*Processing Capacity Constraints.* The total ore processed during each period should be within the predefined upper and lower processing limits:

$$\begin{aligned} \sum_{i=1}^N o_i * w_i * x_{it} &\geq \underline{O}^t, \quad \forall t = 1, 2 \dots T, \\ \sum_{i=1}^N o_i * w_i * x_{it} &\leq \overline{O}^t, \quad \forall t = 1, 2 \dots T. \end{aligned} \quad (6)$$

$\underline{Q}^t, \overline{Q}^t$  represent the lower and upper limits of the available processing capacity in period  $t$ , respectively (*constant*).  $o_i$  is a constant indicator; its value is equal to 1 if block  $i$  is categorized as ore and 0 otherwise.

In a typical open pit mine the UPL may contain thousands to millions of blocks that have to be scheduled over a time horizon typically ranging from 5 to 30 periods or may be more; the resulting integer formulation may contain thousands to millions of integer variables and constraints, which may be extremely difficult and expensive to solve. For instance, defining production schedule for a small open pit mine containing 10000 blocks which needed to be scheduled over 10 periods may require 1,000,00 binary decision variables which is beyond the capabilities of currently available commercial software on the current hardware.

### 3. Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is a stochastic population based optimization technique first proposed by Kennedy and Eberhart in 1995 [20, 21]. PSO is a nature inspired algorithm. It is based on the social interaction of individuals living together in groups, for example, bird flock, fish schools, animal herds, and so forth. PSO algorithm performs the search process by using a population (swarm) of individuals (particles). Each individual (particle) is a potential solution to the optimization problem. At the start a random starting position and random velocity are assigned to each particle of the swarm. The velocity and position of the individual particles are then iteratively adjusted using the following equations for finding better positions in the search space:

$$\begin{aligned}\vec{v}_{i,t} &= \omega \vec{v}_{i,t-1} + c_1 \vec{r}_1 \cdot (\vec{p}_{i,t-1} - \vec{x}_{i,t-1}) + c_2 \vec{r}_2 \cdot (\vec{g}_{t-1} - \vec{x}_{i,t-1}), \\ \vec{x}_{i,t} &= \vec{x}_{i,t-1} + \vec{v}_{i,t},\end{aligned}\quad (7)$$

where

$\vec{x}_{i,t}, \vec{x}_{i,t-1}$  = position of the particle  $i$  during current and previous iterations, respectively,

$\vec{v}_{i,t}, \vec{v}_{i,t-1}$  = velocity of the particle  $i$  during current and previous iterations, respectively,

$\vec{p}_{i,t}$  = personal best position experienced by particle  $i$  until iteration  $t$ ,

$\vec{g}_t$  = global best position discovered by the population until iteration  $t$ ,

$\omega$  = inertia weight used to control the contribution of the particle's previous velocity,

$c_1, c_2$  = acceleration coefficients used to control the influence of cognitive and social terms on the particle's current velocity,

$\vec{r}_1, \vec{r}_2$  = vectors of uniform random numbers between 0 and 1.

The flow chart of the standard PSO algorithm is shown in Figure 1.

**3.1. Guaranteed Convergence PSO (GCPSO).** This modified variant of the PSO algorithm was introduced by Van den Bergh to counter the problem of premature convergence to solutions that are not guaranteed to be the local optimum. The following modified equation was proposed to update the velocity of the best particle of the population or a certain neighborhood:

$$\vec{v}_{b,t+1} = -\vec{x}_{b,t} + \vec{g}_t + \omega \vec{v}_{b,t} + \rho_t (1 - 2\vec{r}_t), \quad (8)$$

where  $b$  represent the index of the best particle and  $\vec{r}_t \in [0, 1]$  is a vector of uniform random numbers.  $\rho$  is a scaling factor used to perform a random search around the global best position; its value is modified during the iterative process using an adaptive strategy [22].

### 4. Particle Swarm Optimization Algorithm for Open Pit Mine Production Scheduling Problem

In this paper the continuous version of particle swarm algorithm and guaranteed convergence PSO algorithm (GCPSO) [23] (with a few modifications) has been applied to open pit mine production scheduling problem. There are two main reasons for using the continuous version of the said algorithms: firstly the open pit mine may contain thousands to millions of blocks inside the ultimate pit limit; making the scheduling decisions on the blocks level may be computationally expensive. Secondly the blocks in the same column of a block model are stacked on top of each other that defines a precedence relationship among these blocks which dictates that the block scheduling problem can be turned into optimum depth determination problem which makes the PSO calculations much faster and the implementation simple [24]. The flow chart of the proposed procedure for applying PSO algorithm to the open pit mine scheduling problem is shown in Figure 4 and will be discussed in more detail in the following sections.

**4.1. Initial Solutions.** To generate an initial population of random feasible solutions a sequential heuristic procedure has been used. The process starts by generating a list of the blocks that are available to be mined in the current period as their predecessor blocks are either already mined or due to their position in the block model they do not have one. A block is selected at random from the list and assigned to the current period and the list is updated again. The process continues for the current period until the mining and processing capacities are satisfied for it in the average sense before moving on to the next period. Mining and processing capacity constraints are handled as hard and soft constraints, respectively, during this process. The process stops when either the free blocks list is empty or the number of periods is finished for the current solution. To further diversify the generated solutions the annual mining capacity per period for each solution is chosen at random from the interval between average and maximum allowed mining capacity.

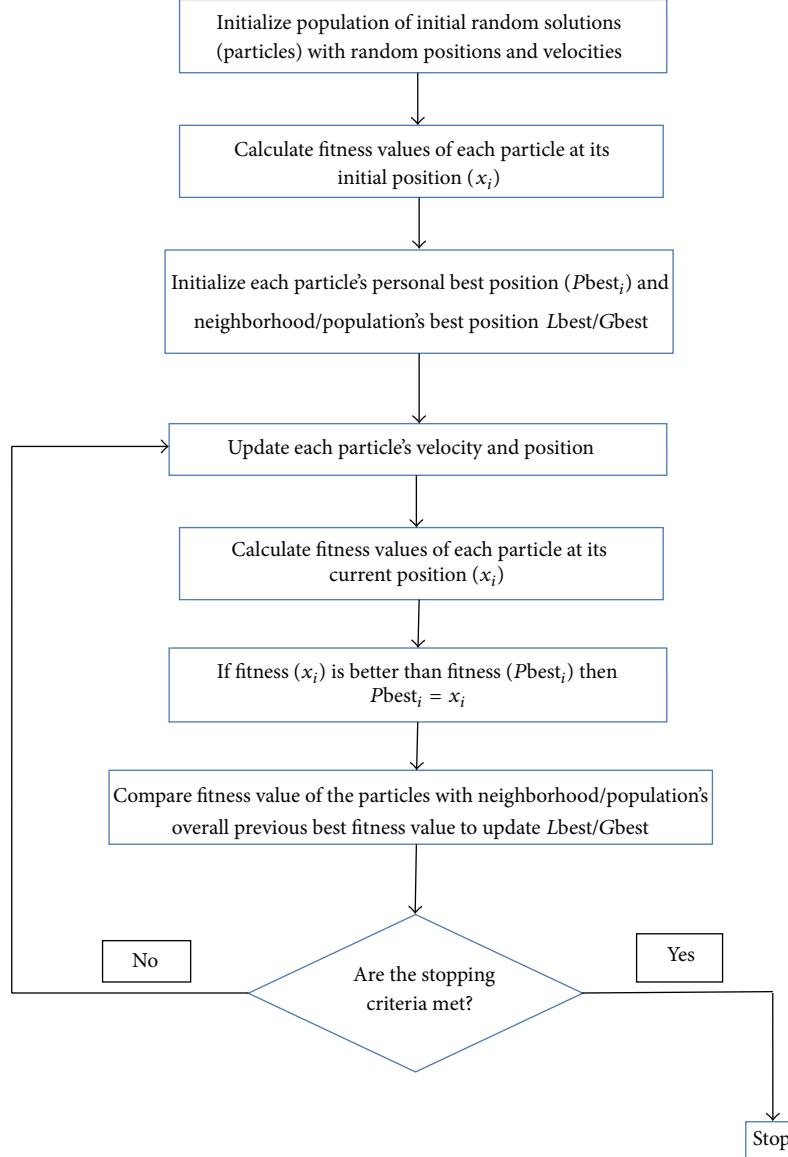


FIGURE 1: Flow diagram illustrating the standard particle swarm optimization (PSO) algorithm.

**4.2. Solution Encoding.** A solution encoding scheme is being used in an effort to apply the continuous version of particle swarm optimization algorithm to open pit production scheduling problem. The proposed encoding scheme determines the deepest block along a certain column to be mined in a certain period. This value (depth) is assigned to a variable corresponding to that column and period. The values of these depths variables are then updated using the standard PSO or GCPSO algorithm in each iteration in an effort to find better solutions.

**4.3. Back Transform.** After each iteration a back transform scheme is used to determine the period to which a block is being assigned. The procedure takes the depth variable for a certain column and period and determines all the blocks that are lying above that depth and below the depth defined by the PSO algorithm for the same column for the prior period.

Obviously there is no upper limit for the first period. This process progresses from the first to the last period. During the optimization process it is ensured that depth for the successive periods along a column does not cross each other.

**4.4. Constraint Handling.** Particle swarm algorithm like other evolutionary algorithms does not have an explicit mechanism for constraint handling. Considering the special structure of the constraints the following two techniques have been used for constraint handling in this study.

**4.4.1. Solution Normalization.** Due to the one-dimensional nature of the PSO algorithm, the mining depths along each column for each period are determined independently from each other which may result in an infeasible solution in terms of the required slope angles as shown in Figure 2. To turn this

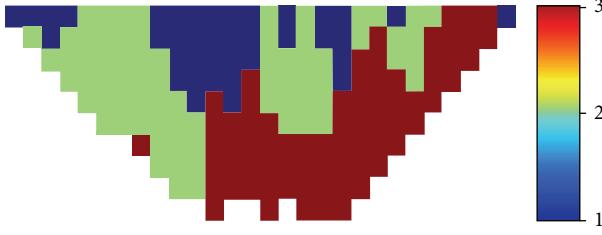


FIGURE 2: A three-period infeasible solution (in terms of the required slope angles) generated by the proposed PSO algorithm during the iterative process.

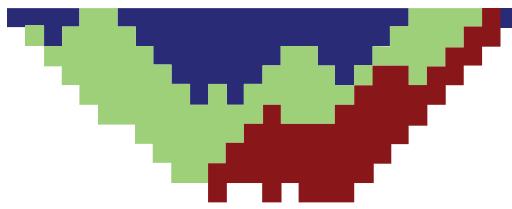


FIGURE 3: Feasible solution in terms of the required slope angles ( $45^\circ$  in this case) after applying the normalization procedure.

back transformed infeasible solution into a feasible solution, a so-called normalization technique is being used.

**4.4.2. Penalty Method for Capacity Constraint Violation.** A constant penalty method [25] is used to deal with the mining and processing capacity constraints, where a constant penalty is added to the objective function for per ton violation of the capacity constraints, to decrease the quality of the infeasible solutions. The value of the penalty is problem dependent and is obtained for each problem using trial and error procedure.

## 5. Numerical Results

The proposed procedure for applying PSO algorithm to production scheduling problem of open pit mines has been implemented using Microsoft Visual studio 2010 (C++) programming environment. The capabilities and the efficiency of different variants of the PSO will be checked using this program. All the numerical experiments have been completed on AMD Phenom IIX 4 945 (3 GHz) and 4 GB Ram running under windows 7.

Two different data sets describing two different hypothetical copper deposits have been used to conduct the numerical experiments. The process started by first dividing the blocks into two categories, that is, ore and waste blocks, using a predefined fixed cutoff grade strategy followed by the definition of the economic block model. The ultimate pit limits for both the deposits were determined by solving the ultimate pit limit problems (as described in Section 1). The required slope angles were assumed to be  $45^\circ$  in all directions (Figure 3). The length of each scheduling period was assumed to be 1 year. The upper and lower limits for processing and mining capacity were set to be within  $\pm 15\%$

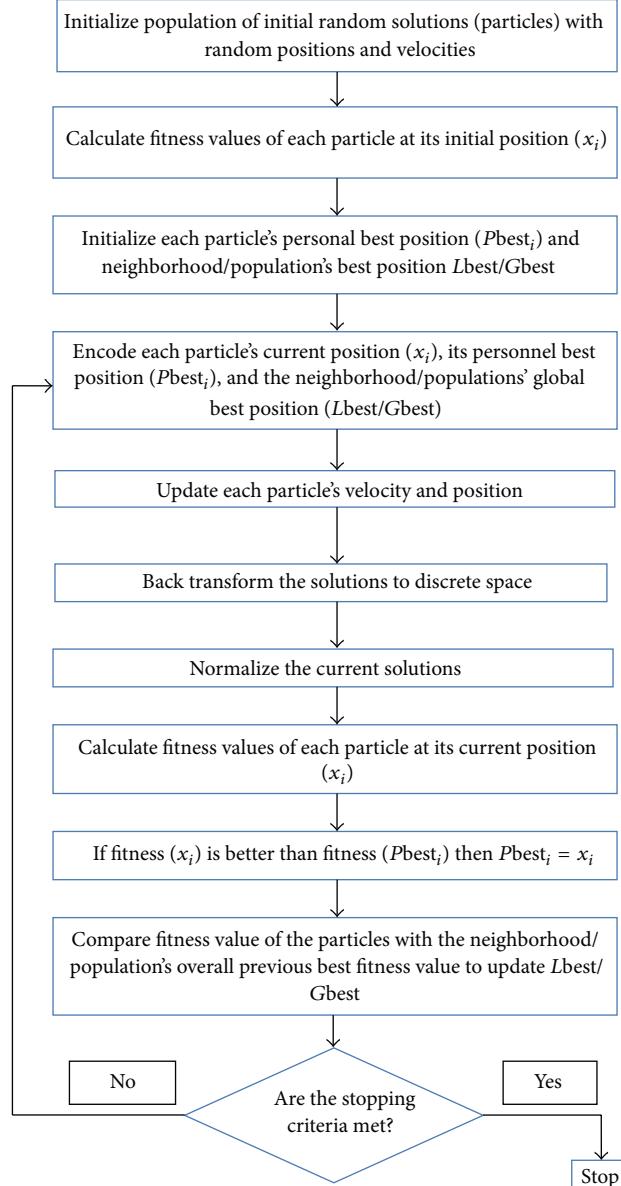


FIGURE 4: Flow diagram illustrating the proposed procedure of applying PSO algorithm (with the modifications) to the open pit mine production scheduling problem.

of the average available quantity of ore and rock within the UPL for each period of the scheduling horizon, respectively. The discount rate was assumed to be 8% per year. The integer programming formulation of the production scheduling problem (as described in Section 2) using above-mentioned data sets and parameters was solved using commercial solver CPLEX; this solution was used as a benchmark to assess the performance of the different variants of the PSO algorithm in terms of computational time and solution quality. General information of the blocks lying inside the ultimate pit limits and about the solutions found by CPLEX is given in Table 1.

The efficiency of the following three different variants of the PSO and GCPSO algorithm has been checked during the numerical experiments.

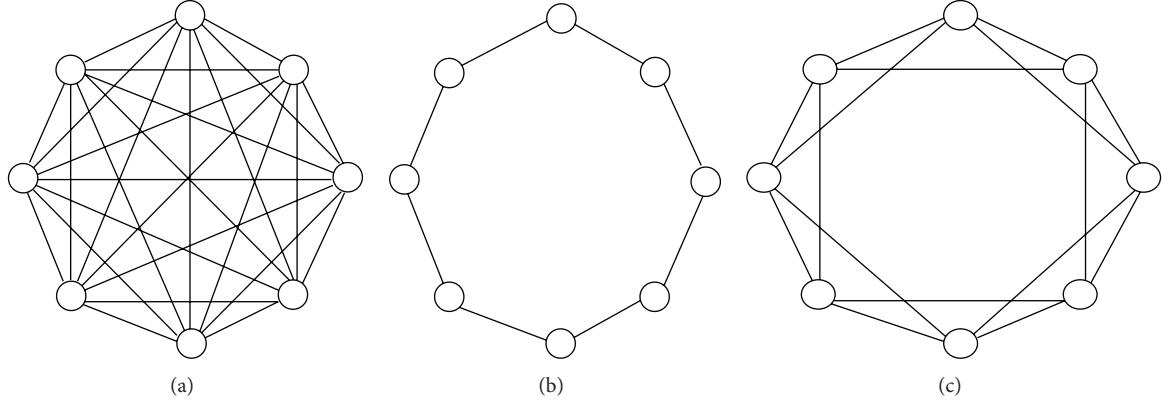
FIGURE 5: Topologies Gbest (a), Lbest with  $k = 2$  (b) and  $k = 4$  (c).

TABLE 1: General information about the blocks within UPL and optimization results.

Problem number	Number of blocks	Block weight (tons)	Block size (m)	Number of periods	Optimality gap (%)	CPU time (minutes)
1	10120	5715	$10 \times 10 \times 10$	5	12.61	12920
2	7836	5715	$10 \times 10 \times 10$	4	4.80	6315

*Global (Gbest).* It is the one where each particle is influenced by the best particle of the entire population.

*Local (Lbest).* Each particle is only influenced by the best particle of its  $K$  immediate neighbours, where  $K$  represents here the total number of indexed based neighbours of each particle of the population.

*Multistart.* Considering the fast convergence behaviour of the global variant of the PSO/GCPSO algorithm a Multistart start strategy has been applied as a diversification strategy, where the particles are reinitialized to their initial position whenever the algorithm does not show any improvement in Gbest value for a certain number of iterations (Figure 5).

During all the experiments the values of the inertia weight ( $\omega$ ) were set to 0.7298 and the values of the acceleration coefficients ( $c_1$  and  $c_2$ ) were fixed to 1.49445 as proposed by Clerc and Kennedy in [26]. After conducting a series of experiments a population of 50 particles generated using the sequential heuristic procedure (described in Section 4.1) were found to be working well for both the problems. The performance of all the variants of the PSO algorithm was found out to be greatly affected by the numerical values of the penalties used during the iterative process. Using relatively higher values was causing the algorithm to get trapped in a local optimum without exploring better solution regions while on the other hand using very low penalties was causing excessive violation of the production capacity constraints. To deal with this problem the following two different sets of penalties have been used during the optimization process.

*Algorithmic Penalties.* These penalties were used during the iterative process for getting acceptable results in terms of the quality of the final solution produced. Their values are data

dependent and needed to be determined using trial and error procedure. The appropriate values were found to be 7 and 9 \$/ton for problems 1 and 2, respectively.

*Actual Penalties.* These are data independent penalties and were used to calculate fitness value of the final solution produced at the end of the iterative process.

During all the experiments, the maximum number of iterations was used as the termination criteria. The appropriate values were determined for each problem independently from each other after observing the convergence behavior of different variants of the PSO algorithm when applied to them (as shown in Figure 6) and were fixed to 30000 and 9000 for problems 1 and 2, respectively. These same values will be used in the case of GCPSO as well. The chances of getting better positions in the consecutive iteration were found out to be quite low so instead of using the adaptive strategy for updating the  $\rho$  value during the iterative process a constant value has been used. A value equal to 5 was found out to be working well for the problems under consideration. While applying the local variant of GCPSO algorithm the positions of the local and global best particle are updated using the GCPSO update equation once separated from the rest of the particles to avoid updating their position multiple times in the same iteration. Due to the stochastic nature of the PSO algorithm, each problem was solved 10 times using each variant of the PSO algorithm. The average relative % Gap between best solutions generated by the different variants of the PSO/GCPSO algorithm, that is,  $Z_{\text{PSO}}$ , and the optimal solution found by CPLEX, that is,  $Z_{\text{CPLEX}}$ , is calculated using the following equation and is reported in Tables 2, 3, 4, and 5:

$$\% \text{ Gap} = \frac{Z_{\text{CPLEX}} - Z_{\text{PSO}}}{Z_{\text{PSO}}} * 100. \quad (9)$$

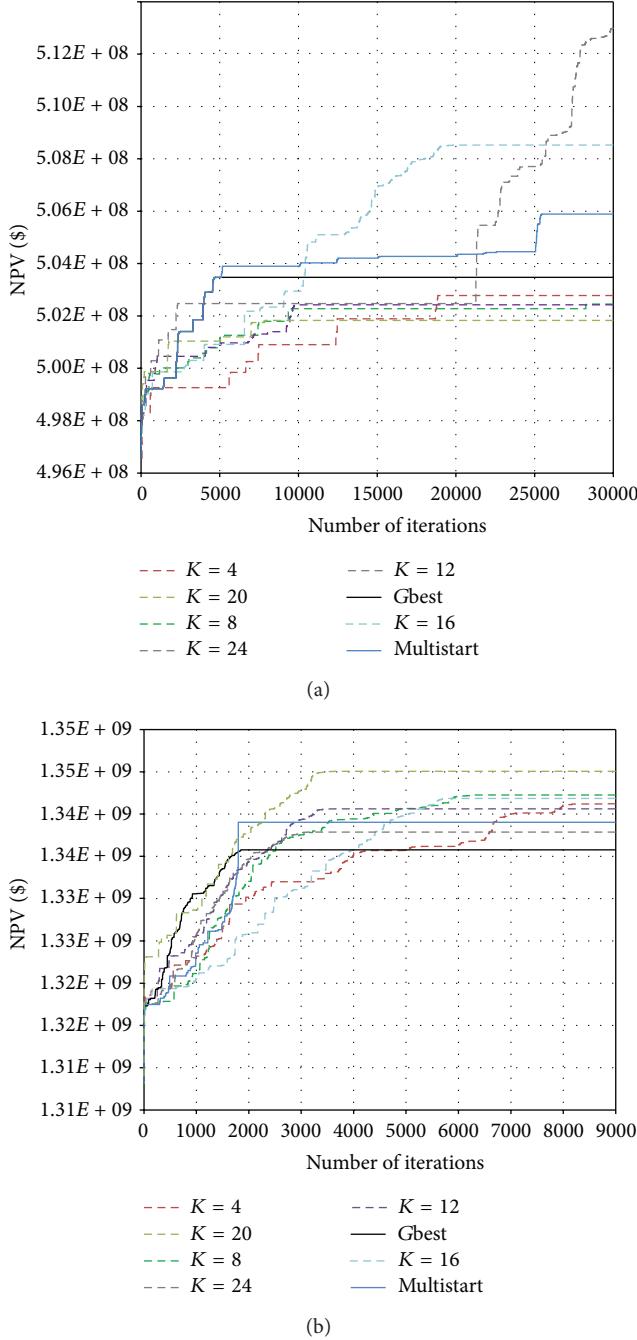


FIGURE 6: Convergence behaviour of the different variants of the PSO algorithm during a single run when applied to problem 1 (a) and problem 2 (b).

The global variant of the PSO algorithm is the one where each particle is influenced by the best particle of the entire population and is considered to be more susceptible to get trapped in a local optimum position and has shown similar kind of convergence behaviour which can be observed in Figure 6. On the other hand the local variants show relatively slower rate of convergence with different neighbourhood sizes. The Multistart strategy proved to be a better option to overcome the premature convergence problem of the global

TABLE 2: Numerical results of different variants of the PSO algorithm in the case of problem 1.

Local PSO	% GAP		CPU time (minutes)
	Mean	Standard deviation	
$K = 4$	5.46	0.36	132.36
$K = 8$	5.01	0.51	129.96
$K = 12$	4.84	0.50	131.77
$K = 16$	4.67	0.67	132.31
$K = 20$	5.03	0.63	134.83
$K = 24$	4.67	0.61	134.25
Global PSO	5.16	0.51	124.34
Multistart PSO	4.43	0.59	124.09

TABLE 3: Numerical results of different variants of the GCPSO algorithm in the case of problem 1.

Local GCPSO	% GAP		CPU time (minutes)
	Mean	Standard deviation	
$K = 4$	5.84	0.17	138.83
$K = 8$	5.36	0.27	143.09
$K = 12$	5.53	0.31	145.75
$K = 16$	5.07	0.41	145.34
$K = 20$	5.38	0.44	145.55
$K = 24$	5.44	0.31	145.79
Global GCPSO	5.15	0.51	122.40
Multistart GCPSO	4.34	0.50	118.19

TABLE 4: Numerical results of different variants of the PSO algorithm in the case of problem 2.

Local PSO	% GAP		CPU time (minutes)
	Mean	Standard deviation	
$K = 4$	2.00	0.24	22.55
$K = 8$	1.95	0.30	21.25
$K = 12$	1.82	0.13	21.31
$K = 16$	1.92	0.16	21.60
$K = 20$	1.90	0.20	21.76
$K = 24$	1.99	0.25	21.86
Global PSO	2.23	0.22	22.46
Multistart PSO	2.06	0.27	24.26

variants of both the PSO and GCPSO algorithms and on average has produced better solution with smaller relative gap as can be seen in Tables 2, 3, 4, and 5. The standard deviation of % Gap in almost all the cases has very small values showing the robustness of the procedure. Generally the average relative gap tends to increase with problem size. The computation time required by the algorithm depends on the factors defining and increasing the size of the required computations to be performed before the termination criteria are met such as population's size, number of blocks, and number of periods. The computation time increases linearly

TABLE 5: Numerical results of different variants of the GCPSO algorithm in the case of problem 2.

Local GCPSO	% GAP		CPU time (minutes)
	Mean	Standard deviation	
$K = 4$	3.28	0.26	22.29
$K = 8$	2.96	0.39	22.72
$K = 12$	3.09	0.27	23.17
$K = 16$	2.70	0.50	23.50
$K = 20$	2.63	0.63	23.53
$K = 24$	2.28	0.36	23.94
Global GCPSO	2.67	0.47	20.32
Multistart GCPSO	2.70	0.62	20.12

with the number of iterations for a specific problem. The proposed procedure in all the cases can produce relatively better quality results in relatively shorter period of time in comparison to the exact optimization algorithms implied by CPLEX.

## 6. Conclusion

This paper presents a framework for applying a computationally efficient population based metaheuristic technique known as particle swarm optimization algorithm to the long term production scheduling problem of the open pit mines. Instead of making the scheduling decision on the block level the proposed approach turns the problem into depth finding problem and then applies the continuous version of the PSO algorithm and its different variants to find the optimum or near to optimum solution. A greedy and fast heuristic technique was used to generate the initial population of random feasible solutions which were transformed over several generations using particle swam procedure to get optimum or near to optimum solution of the production scheduling problem. By making comparison with the results obtained using CPLEX in terms of computational time and solution quality it was learned that the proposed procedure can produce better quality solutions in relatively shorter period of time with smaller % Gap and standard deviation showing the robustness of the procedure. The proposed procedure is more structured and quite flexible and can easily accommodate additional constraints, variable slopes angles, and grade and market uncertainties.

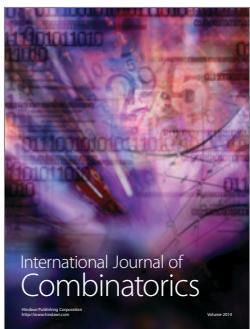
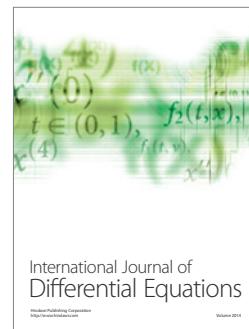
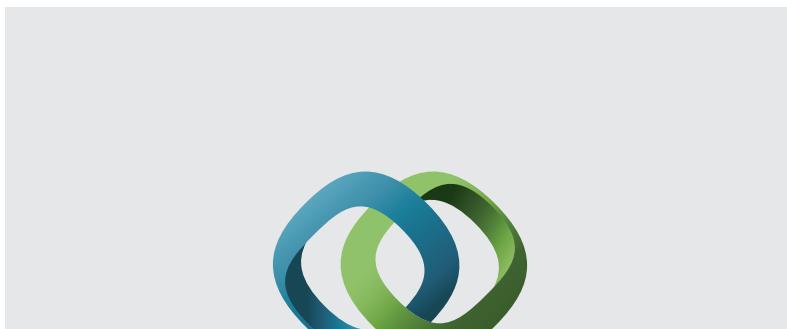
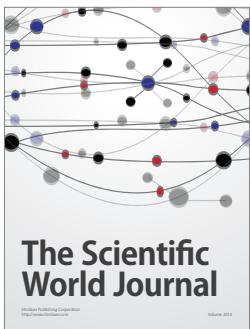
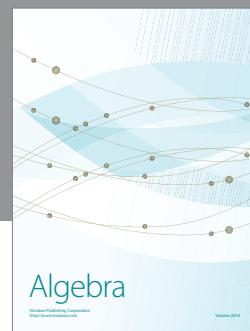
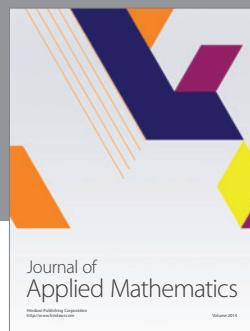
## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] H. Lerchs and I. F. Grossmann, "Optimum design of open pit mines," *Canadian Institute of Mining Transactions*, vol. 68, pp. 17–24, 1965.
- [2] T. B. Johnson, *Optimum Open Pit Mine Production Scheduling*, University of California, Berkeley, Calif, USA, 1968.
- [3] K. Dagdelen and T. B. Johnson, "Optimum open pit mine production scheduling by Lagrangian parameterization," in *Proceedings of the 19th International Symposium Application of Computers and Operations Research in the Mineral Industry (APCOM '86)*, pp. 127–142, Littleton, Colo, USA, 1986.
- [4] L. Caccetta, P. Kelsey, and L. Giannini, "Open pit mine production scheduling," in *Proceedings of 3rd Regional APCOM International Symposium*, Series 5/98, The Australasian Institute of Mining and Metallurgy, Kalgoorlie, Australia, 1998.
- [5] J. Whittle, *Four-X User Manual*, Whittle Programming, Melbourne, Australia, 1998.
- [6] M. E. Gershon, "Mine scheduling optimization with mixed integer programming," *Mining Engineering*, vol. 35, no. 4, pp. 351–354, 1983.
- [7] L. Caccetta and S. P. Hill, "An application of branch and cut to open pit mine scheduling," *Journal of Global Optimization*, vol. 27, no. 2-3, pp. 349–365, 2003.
- [8] A. H. Onur and P. A. Dowd, "Open-pit optimization. Part 2. Production scheduling and inclusion of roadways," *Transactions of the Institution of Mining and Metallurgy Section A: Mining Technology*, vol. 102, pp. 105–113, 1993.
- [9] Q. Wang, "Long-term open-pit production scheduling through dynamic phase-bench sequencing," *Transactions of the Institution of Mining and Metallurgy Section A*, vol. 105, pp. A99–A104, 1996.
- [10] S. Ramazan, "The new fundamental tree algorithm for production scheduling of open pit mines," *European Journal of Operational Research*, vol. 177, no. 2, pp. 1153–1166, 2006.
- [11] B. Denby and D. Schofield, "Open-pit design and scheduling by use of genetic algorithms," *Transactions of the Institution of Mining and Metallurgy, Section A: Mining Technology*, vol. 103, pp. A21–A26, 1994.
- [12] B. Denby and D. Schofield, "Genetic algorithms for open pit scheduling-extension into 3-dimention," in *Proceedings of the Mine Planning and Equipment Selection Conference (MPES '96)*, pp. 177–185, Sao Paulo, Brazil, 1996.
- [13] B. Denby, D. Schofield, and T. Surme, "Genetic algorithms for flexible scheduling of open pit operations," in *Proceedings of the 27th Application of Computer in Mineral Industry Symposium (APCOM '98)*, pp. 473–483, London, UK, 1998.
- [14] M. Kumral and P. A. Dowd, "Short-term mine production scheduling for industrial minerals using multi-objective simulated annealing," in *Proceedings of the 30th Application of Computer in Mineral Industry Symposium (APCOM '02)*, pp. 731–742, Fairbanks, Alaska, USA, 2002.
- [15] M. Kumral and P. A. Dowd, "A simulated annealing approach to mine production scheduling," *Journal of the Operational Research Society*, vol. 56, no. 8, pp. 922–930, 2005.
- [16] J. Sattarvand and C. Niemann-Delius, *Long Term Open Pit Planning by Ant Colony Optimization*, Institut für Bergbaukunde III, RWTH Aachen University, 2009.
- [17] J. Sattarvand and C. Niemann-Delius, "A new metaheuristic algorithm for long-term open-pit production planning / Nowy meta-heurystyczny algorytm wspomagający długoterminowe planowanie produkcji w kopalni odkrywkowej," *Archives of Mining Sciences*, vol. 58, no. 1, pp. 107–118, 2013.
- [18] J. A. Ferland, J. Amaya, and M. S. Djuimo, "Application of a particle swarm algorithm to the capacitated open pit mining problem," in *Autonomous Robots and Agents*, S. Mukhopadhyay and G. Sen Gupta, Eds., pp. 127–133, Springer, Berlin, Germany, 2007.

- [19] M. P. Gaupp, *Methods for Improving the Tractability of the Block Sequencing Problem for Open Pit Mining*, Business and Economics Division, Colorado School of Mines, 2008.
- [20] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, October 1995.
- [21] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [22] F. van den Bergh, *An Analysis of Particle Swarm Optimizers*, Faculty of Natural and Agricultural Science, University of Pretoria, 2001.
- [23] F. van den Bergh, *An Analysis of Particle Swarm Optimizers*, Department of Computer Science, University of Pretoria, 2002.
- [24] P. Angeline, “Evolutionary optimization versus particle swarm optimization: philosophy and performance differences,” in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., vol. 1447 of *Lecture Notes in Computer Science*, pp. 601–610, Springer, Berlin, Germany, 1998.
- [25] Z. Michalewicz and M. Schoenauer, “Evolutionary algorithms for constrained parameter optimization problems,” *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1996.
- [26] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.



Submit your manuscripts at  
<http://www.hindawi.com>

