PSO Algorithm for Block Sequencing Problem in Open Pit Mining

Wenbo Liu[1, a], Hongyi Zhang[1, b]

[1]Liaoning Key Laboratory of Manufacturing System and Logistics, The Logistics Institute, Northeastern University, Shenyang, 110819, China

[a]email: wenbo-315@163.com, [b]email:735225202@qq.com

Keywords: Mining, Particle Swarm Optimization algorithm, CPLEX

Abstract. Iron ore is the main raw material of the production in iron and steel enterprises in China. It is a non-renewable resource and with limited reserves. In this paper, in combination with the practical situation of the mine, an integer linear programming model of a strip mining block scheduling problem is established with the scientific, reasonable, and economical principle. The maximization of mining profit is set as the goal under the precedence constraints and production capacity constraints. This paper designs an improved particle swarm optimization algorithm to solve the problem and compares with results with CPLEX optimizing software. According to the results of different scale experiment, the improved particle swarm optimization algorithm has better performance.

Introduction

In the iron and steel enterprises, various raw materials are essential to produce iron and steel. Since the prices of imported raw materials are increasing significantly, optimizing the domestic mining is very promising and profitable. The logistics costs of mining are mainly concentrated in mining production, inventory management and vehicle scheduling, etc.

The optimization of mining production can excavate reasonable forecast of mineral deposits in the space, and moreover, achieve the goal of optimizing the mining sequence and maximizing the profit while satisfying the production requirements and resources limits, such as geological data, mining equipment, technical constraints.

In recent years, a variety of modeling and solution methods have been proposed in the research of the domestic mining problem, such as linear programming, integer programming, dynamic programming, interactive dynamic optimization method, the directed graph simulation method, genetic algorithm and heuristic algorithm. According to actual conditions of the mining, this paper builds the sequence open mining model and uses particle swarm optimization algorithm to solve the problem. And comparison with CPLEX is conducted.

As shown in figure 1, the most commonly used model of mining ore deposits is 3D block model. 3D block model is the discrete model which divides ore deposits into cell blocks. With the application and constantly improvement of computer instrument in mining, 3D block model has been used more and more widely in the world. 3D block model is widely used not only to calculate the ore grade and quantity, but also to open pit mining optimization. In fact, many optimization methods appeared due to the introduction of the 3D block model[1], as well as most of ongoing research. Given this situation, our problem, i.e. mining production scheduling problem, can be also called as block sequencing problem[2].

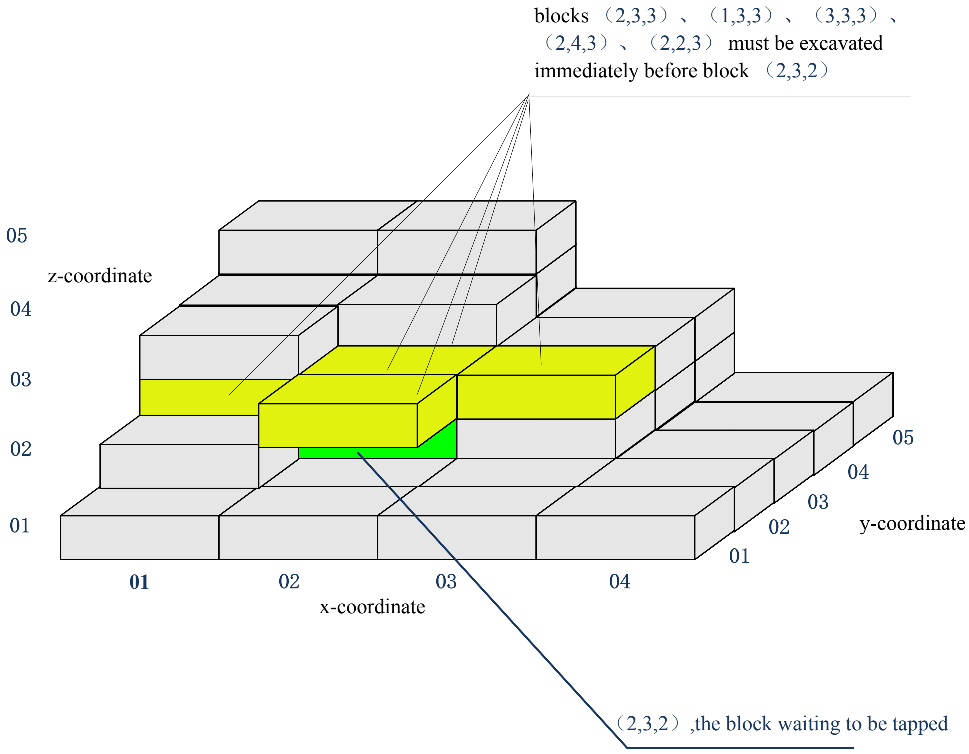
blocks （2,3,3）、（1,3,3）、（3,3,3）、（2,4,3）、（2,2,3） must be excavated immediately before block （2,3,2）
05
z-coordinate
04
03
02
01
01
02
03
04
x-coordinate
05
04
03
02
01
y-coordinate
（2,3,2） ,the block waiting to be tapped

Fig. 1 Three-dimensional block model diagram

Mathematical model

Model parameter

b: ore block index.
B: set of all ore blocks.
t: time period.
T: set of periods within the horizon.
B$_b$: set of blocks that must be excavated immediately before block *b*.
v_{bt} : net value associated with the extraction of block *b* in period *t*.
C_b: consumption of resource associated with the extraction of block *b*.
C_{min}: minimum resource limit in any period.
C_{max}: maximum resource limit in any period.

Decision Variable

y_{bt} : 1 if block b is extracted in period t, 0 otherwise.

The model

$$\max \sum_{t\in T}\sum_{b\in B} v_{bt}y_{bt} \tag{1}$$

$$\sum_{t\in T} y_{bt} \le 1 \quad \forall b \tag{2}$$

$$\sum_{b\in B} C_b y_{bt} \ge C_{\min} \quad \forall t \tag{3}$$

$$\sum_{b\in B} C_b y_{bt} \le C_{\max} \quad \forall t \tag{4}$$

$$y_{bt} \le \sum_{\tau=1}^{t} y_{b'\tau} \quad \forall t, b \in B, b' \in B_b \tag{5}$$

$$y_{bt} \in \{\ 0,1\ \} \quad \forall t, b \tag{6}$$

The objective function (1) means the maximum net present value of the exploitation ore deposit. Where there is 0-1 variables, if = 1, ore block has been mined at the moment. Otherwise it hasn’t been mined. Constraints (2) means during the whole mining horizon *T*, any ore block can only be extracted in at most one time period. Constraint (3) and (4) indicate the minimum and maximum capacity limits in each time period, respectively. Constraints (5) ensure the mining sequence requirement is satisfied, which is the key process requirement. We must obey the sloping requirements and sequencing constraints to ensure that the pit walls do not collapse. A commonly assumed slope requirement is 45◦, meaning that in order to mine block (2,3,2) in Fig.1, blocks (2,3,3), (1,3,3), (3,3,3), (2,4,3) and (2,2,3) must be mined first.

The Particle Swarm Optimization algorithm

The Particle Swarm Optimization

The particle swarm optimization algorithm is a kind of evolutionary computation technique based on population and is proposed by Eberhart and Kennedy in 1995. The particle swarm optimization algorithm is used to imitate the cluster behavior of insects, birds, fishes and so on. These swarms find food in a cooperative way that each member of the swarm changes its search pattern continually according to learning experience of itself and other members[3].

Kennedy and Eberhart proposed the particle swarm optimization algorithm from two main aspects: One is the evolutionary computation. The particle swarm optimization algorithm adopts the method of population search. It is similar with other evolutionary algorithms on this point, which makes it can search more in feasible solution space at the same time, the difference with other evolutionary algorithms is that it regards each individual as a particle without quality and volume particles in optimization space, flying in the search space at a certain speed. Through learning and adapting to the environment and flight experience of the individual and swarm comprehensive analysis to dynamically adjust the speed. Second, artificial life, namely the life characteristics of artificial systems, it is mainly simulated by computer programming[4].

The initialized algorithm is a set of random values, through the particles in the solution space to follow individual extremum and global extremum to search the optimal solution of the problem.

Algorithm takes every solution of the optimization problem solution space as a bird in the search space, that is, a "particle", each particle has a location (a vector) to determine the particle's position in the search space, each particle has a speed (dimension of a vector with location) to determine the direction and distance of the flying particles, all the particles have fitness values determined by the objective function. The fitness of each particle in each iteration is an individual extremum, all particles in each iteration jointly own a global extreme value, the particle will follow individual extremum and global extremum search in the solution space. The composition of particles in PSO algorithm is the basic unit to represent the solution space of a candidate solution[5].

A swarm is composed of *m* particles in D dimensional search space and flies at a certain speed. During searching, each particle will consider the best point in his past search and also the best points of other particles in the neighborhood in their past search history.

The position of particle *i* is expressed as: $x_i = \left(x_{i1}, x_{i2}, \dots, x_{iD}\right)$.

The speed of particle *i* is expressed as: $v_i = \left(v_{i1}, v_{i2}, \dots, v_{iD}\right), 1 \le i \le m, 1 \le d \le D$.

The historical best point of particle *i* is expressed as : $p_i = \left(p_{i1}, p_{i2}, \dots, p_{iD}\right)$.

The best point of all particles within the swarms (or in the neighborhood) is expressed as : $p_g = \left(p_{g1}, p_{g2}, \dots, p_{gD}\right)$.

In general, particle's position and speed are values in continuous space of real number. For continuous problem, update formula of particle's position and speed is as follows:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 \xi \left(p_{id}^{k} - x_{id}^{k}\right) + c_2 \eta \left(p_{gd}^{k} - x_{id}^{k}\right) \tag{7}$$

$$\begin{cases} v_{id}^{k} = v_{\max} \quad if \quad v_{id}^{k} \ge v_{\max} \\ v_{id}^{k} = -v_{\max} \quad if \quad v_{id}^{k} \le v_{\max} \end{cases} \tag{8}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{9}$$

Where ω is inertia weight, c_1 and c_2 are called Learning Factor, generally they are positive constants. Learning Factors make particles have self summary and the ability to learn from excellent individuals in swarm, thus to be near their own historical best points and the global best point within the swarms (or in the neighborhood), c_1 and c_2 usually equal to 2. $\xi,\eta \in U[0-1]$, the pseudo-random number which submit to uniform distribution in the range of [0,1]. In the above formula, the speed of particle v_{id}^k has maximum value limit. If the current speed acceleration of particles makes a certain velocity components v_{id}^k greater than the maximum speed $v_{\max}$, the speed is restricted to maximum speed $v_{\max}$, which determines the search accuracy of particles in the solution space. If $v_{\max}$ is too high, particles may fly over the optimal solution, if $v_{\max}$ is too small, particles easily fall in local search space and cannot search globally.

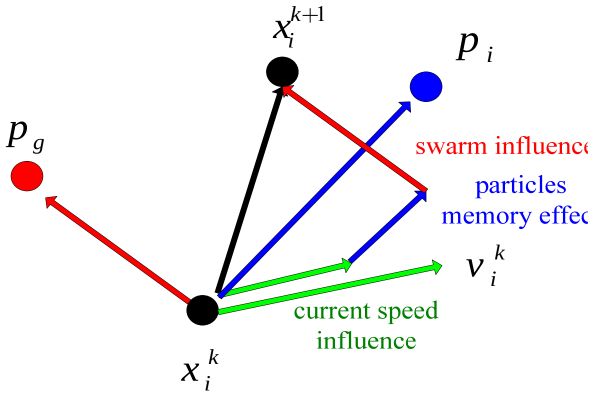

Fig. 2 A schematic of particle iteration

The basic Particle Swarm Algorithm

1) Setting of algorithm parameters

Particle swarm algorithm has experienced a lot of adjustment and correction until now. Many researchers made a great deal of analysis and experiments on the choice of parameters and its effect on the performance of algorithm, which has laid a solid foundation for the particle swarm optimization algorithm theory and application research[6].

Algorithm parameter settings are as follows:

(1) The swarm size *m*

m is the integer parameter. When the value m is small, solving process may fall into local optimum, but if the swarm size is set too big, the computing time would increase greatly,convergence speed will be slow, and when the number of population grows to a certain level, the growth of size will not be helpful to the improvement of solution, thus will have greatly influence on the performance of the algorithm. In this paper, the swarm size is set to 5.

(2) The inertia weight ω

ω is inertia weight. The performance of PSO mainly depends on balance between algorithm exploration ability and development ability. For the PSO algorithm, the balance of these two kinds of ability is accomplished by adjusting the size of the inertia weight. If the value of ω is greater, the exploring ability of algorithm is stronger, while development capability will be weaken. On the other hand, the development ability will be enhanced, and explore ability will be weaken. By adjusting the size of the inertia weight ω, we can control the current speed's influence on the speed of the next generation, making it a compromise between global search and local search. This is because that, if ω is big, the speed v_{id} will be big, and the particles can fly farther in the solution space to find new domain. While ω is smaller, speed v_{id} will be small to find a better solution in the current space. For the basic Particle Swarm Optimization algorithm ω value is initially set to 1, and time-varying inertia weight will be introduced into the algorithm later.

(3) Maximum speed $v_{\max}$

$v_{\max}$ is maximum speed. The iteration of Particle Swarm Optimization algorithm is carried out through adjusting each particle distances in each dimension. And maximum speed $v_{\max}$ determines the biggest mobile distance of the particle in one iteration thus avoiding the particles in an uncontrolled particle orbit. When $v_{\max}$ is small, development ability strengthens, but easy to fall into local optimum, when $v_{\max}$ is bigger, the ability of exploration will be enhanced, but particle will easily miss the best solution. According to the actual problem, we will set $v_{\max}$ to 2000.

(4) Learning factor c_1, c_2

Learning Factors can help particles do self summary and have the ability of learning from excellent individuals in swarm, thus to get close to the global and local optimum. Specifically, the adjustment of factor c_1 can adjust their learning ability in the proportion of velocity updating formula, while c_2 is weight of the social experience in the role of movement. In this paper, we will set c_1, c_2 to 2.

(5) The number of iterations

The numbers of iterations are set to 50,100 and 1000 for small, medium and large-scale problems, respectively.

2) Encoding

First of all, the set of decision variables y_{bt} (b=1,...,m, t=1,...,n) is expressed in matrix described as follows:

$$y=\begin{bmatrix} y_{11} & .. & .. & y_{1n} \\ .. & .. & .. & .. \\ .. & .. & .. & .. \\ y_{m1} & .. & .. & y_{mn} \end{bmatrix} \qquad （10）$$

The particle coordinates are described using $y=\{y_{ij}\}$ to indicate the position of particle motion in the mn dimensional space, and the speed of movement of the particles can be represented by the matrix $V=\{v_{ij}\}$. According to the table dispatching method, the initial basic feasible solution is generated by random initialization process, meeting the constraints of initial particles.

Initialization process produces a particle $y=\{y_{ij}\}$ satisfying the constraints, which is a matrix containing m+n-1 non-zero elements, representing the convex vertex in the feasible solution space.

In order to solve the optimization problem, as shown in the formula (1), we should design the feasibility conditions which can keep the particle velocity. Based on the problem feasible set and the velocity of particles in the particle swarm optimization algorithm and the characteristics of location updating formula, we can get the new velocity updating formula: $v_{id}^{k+1}=\omega\left(y_{1id}^{k}-y_{2id}^{k}\right)$. So, in order to meet the every step iteration to obtain the constraint condition of particle swarm, we should not only satisfy the constraints of feasible particles, but also need to meet the new initial velocity.

Particle swarm optimization algorithm does not have selection, crossover and mutation operations, therefore the algorithm structure is relatively simple and runs quickly. During the solving process, however, if a particle found a current optimal position, other particles will quickly concentrate to it. If a local optimal point is found, the particle swarm can't search again within the solution space, at this point, the algorithm falls into local optimum, the so-called phenomenon of premature convergence. Experiments show that the particle swarm optimization algorithm premature convergence and global convergence, particle swarm particles will appear at a specific location or a few specific location gathered phenomenon, it mainly depends on the characteristics of the problem itself and the selection of fitness function. The following theoretical analysis is related to this conclusion that the position of the particles in particle swarm optimization algorithm is equivalent to the fitness of each particle. Therefore, studying the fitness status of all particles can tell us the state of the particle swarm [7].

3) Repair strategy of the solution

For each particle swarm in the next generation, its position is calculated using the contemporary particle's position and speed of the next generation, making it easy to result in infeasible solution. Therefore, when calculating the fitness value function, reparation of infeasible solution needs to be done first. The specific strategies are described below[8]:

(1) First, mining precedence constraints should be satisfied to meet for a given deposit. After dividing good ore block, the grade of each ore block is known. To achieve maximum net present values, we need to follow the priorities rules of mining, that is, from top to bottom, from high to low.

(2) Second, mining resources consumption constraints is considered to meet the given precedence constraints to ensure that the resource consumption is within a given limit.

(3) Finally, make sure that each ore block is mined only in one time period and every ore block is continuously mined.

4) The basic process of particle swarm algorithm

Step 1: The position and the speed range of each particle are determined in the initial population.

Step 2: Given the position and the speed range of the particle, initial population is randomly generated.

Step 3: According to the particle position and mapping relation of particle mining sequence, the sorting of mining block is done.

Step 4: Using the production strategy to get mining time, mining sequence of ore block and exploitation value, and evaluate individual particle.

Step 5: Update the best record of each individual particle in a population, and also the global best record of the entire population.

Step 6: Find the best particle in current population, and do the large neighborhood search.

Step 7: Determine whether terminate criteria is achieved, if not, continue to step 8, otherwise, go to step 9.

Step 8: Update the speed and position of particles in the population and go to step 2.

Step 9: Algorithm is terminated, output the global best particle record.

Improved particle swarm algorithm

1) Improvement strategy of particle swarm optimization algorithm

In standard particle swarm optimization algorithm, the determination of the history optimum information is equivalent to an implied selection mechanism. In the neighborhood topology structure, the selection mechanism may need a long time to work. While traditional methods can be selected in the evolutionary algorithm search direction in a better area with reasonable distribution of limited resources. Angeline combined natural selection mechanism with particle Swarm optimization algorithm and proposed a hybrid swarm algorithm (Hybrid Swarm). This paper applies the improved particle swarm optimization algorithm. The hybrid algorithm employs Tournament Selection Method (Tournament Selection Method), in which each particle's adaptive value is compared with other particle's fitness, and then the whole particle swarms are scored from high to low. In the process, the history of the individual optimal value is not considered. Swarm sorting is completed with a swarm of the best half of the current position and velocity to replace the worst half of the position and speed, while retaining the original history and the optimal value of individual memory. So, after each iteration, half of the particles will move into the search space which is related to optimal position, these individuals still retain the original historical information, in order to update the location of the next generation.

The decision variables y_{bt} (b=1,...,m, t=1,...,n) are expressed in matrix form, particle coordinates are expressed in $y=\{y_{ij}\}$, describing movement location of the particle in the mn dimensional space, and the speed of movement of the particles can be represented by the matrix $V=\{v_{ij}\}$. According to the determination of the initial basic feasible solution method random initialization process is used to meet the constraint condition of initial particles.

2) Improved Particle Swarm Optimization algorithm process

Step 1: Initialize all particles y_{ij}, and get a new initial velocity. Set optimal solution initial individual history P_{best} and community history optimal solution g_{best} (in turmoil and location at the same time set).

Step 2: Form a complete production process and evaluate the fitness of each particle value, and its selection.

Step 3: For each particle, compare its fitness value and experience of the best location P_{best}, and test the feasibility of the particle condition, if the particle went well and meet $y_{ij} \geq 0$, the best position P_{best} is set as current.

Step 4: For each particle, compare its fitness value with global experience of the best location, and test the feasibility of the particle condition, if the particle went well and meet the best position is set as a current.

Step 5: Adjust the particle's position and speed according to the velocity updating formula and location updating formula and set up a small positive number $\varepsilon \leq 10^{-6}$. If a particle's speed is less than ε, and remains less than in 20 iterations, then initialize the speed and position.

Step 6: Check whether the maximum number of iterations is met. If the above condition is met, terminate the iteration, otherwise return to step 2.

The experimental results and analysis

Data source mainly includes two parts: the part comes from through the field survey data of mining enterprises, the other part is randomly generated based on the actual data.

Input of the algorithm include: ore body scale, the value of unit ore block, the mining cost of each ore resources consumption, the consumption of mining, the maximum and minimum values of the precedence constraints between different ore block, etc.

The distribution of ore block depends on the size of the ore body, which is the scale of the problem solving. According to the actual circumstance of mine, each field is generally divided into any number of ore block number, as far as possible to ensure uniform distribution of the ore block size.

Calculation results show that, the Particle Swarm Optimization algorithm for this model can well deal with mining block scheduling problems. CPLEX software can't get the optimal solution for large-scale numerical example. For other examples the results of basic Particle Swarm Optimization algorithm are compared with the optimal solution, the average error is within 2%, and computing speed of the particle swarm optimization algorithm is superior to CPLEX software.

Table 1 Result contrast of the algorithms

case	block/period	Basic PSO		Improved PSO		CPLEX	
		Obj	Time/s	Obj	Time/s	Obj	Time/s
1	9/5	129.45	1.42	132.69	1.43	139.63	15.1
2	22/5	568.63	5.25	590.25	5.95	602.53	96.9
3	35/6	751.87	6.72	779.43	6.85	810.95	147.8
4	40/6	863.28	7.91	885.28	7.12	891.73	176.4
5	50/6	1251.74	9.793	1287.6	9.35	1307.83	258.9
6	56/10	1339.27	35.93	1399.622	29.48	1408.37	374.8
7	304/6	5894.1	37.93	5981.8	26.54	6023	1015.74
8	304/10	14686	39.64	14864	27.55	15601.2	1289.57
9	1060/6	39289	209.32	40400	185.16	-	-
10	1060/10	66958	359.28	70635.6	337.37	-	-
11	1980/7	85139	501.56	89875.1	485.27	-	-
12	2880/7	108623	855.37	114653.3	828.6	-	-
13	9400/10	308908	1389.4	328955	1201.6	-	-

Conclusions

In this paper the open pit mining block scheduling problems are described. According to the problem characteristics the integer programming model was constructed. Particle swarm optimization algorithm is proposed to solve the model, and according to the characteristics of the mining sequence problem, the improvement strategy is presented. Through testing the different examples, the experimental results show that the improved particle swarm optimization algorithm performs better, and the solving time is faster than the basic particle swarm optimization algorithm and CPLEX software.

References

[1] Govemey G V R, Sarin S C. Using mathematical program mining to develop optimal overburden transport strategies in a surface coal mining operation [J], Surface Mining Reclamation Environ, 1988, 2(1): 51-58.

[2] Lambert W.B., Newman A.M. Analyzing solutions of the open pit block sequencing problem obtained via Lagrangian techniques, Mining Engineering, 2013, 65(2):47-53

[3] Kennedy J, Eberhart R C. Particle Swarm Optimization [J], IEEE International Conference on Neural Networks, 1995, 4: 1942-1948.

[4] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm [J], Computational Cybernetics and Simulation IEEE International Conference, 1997, 5: 4104-4109.

[5] Goldbarg E F G, Souza G R, Goldbarg M C. Particle Swarm for the traveling salesman problem [J], European Conference on Evolutionary Computation in Combinatorial Optimization, 2006, 3906: 99-110.

[6] Shi Y, Eberhart R C. Empirical study of Particle Swarm Optimization [J], Congressional Evolutionary Computation, 1999, 1945-1950.

[7] Clerc M, Kennedy J. The particle swarm:explosion, stability and convergence in multi-dimensional complex spaces [J], IEEE Transactions on Evolutionary Computation, 2002, 8(1): 58-73.

[8] Clerc M. Discrete particle swarm optimization [J], New Optimization Techniques in Engineering, 2004.