YUN, Q.X., GUO, W.W., CHE, Y.., LU, C.W., and LIAN, M.J. Evolutionary algorithms for the optimization of production planning in underground mines. *Application of Computers and Operations Research in the Minerals Industries*, South African Institute of Mining and Metallurgy, 2003.

Evolutionary algorithms for the optimization of production planning in underground mines

Q.X. YUN, W.W. GUO, Y.F. CHEN, C.W. LU, and M.J. LIAN
Xian University of Architecture and Technology, Xian, Shaanxi Province, Peoples Republic of China

This paper provides a new approach to optimize the production planning in underground mines by means of evolutionary algorithms. Firstly, genetic algorithm is used to reach the global optimization in planning. Then, evolutionary programming is employed to obtain the partial optimization for each year. The application proves that the new approach is reasonable and useful.

Keywords: Evolutionary algorithm, Genetic algorithm, Evolutionary programming, Optimization, Mining engineering.

Introduction

Evolutionary algorithm is a new approach for optimization by means of self-adaptive search. Usually it consists of four kinds of methods: genetic algorithm, genetic programming, evolutionary strategy and evolutionary programming[1]. In essence, all of them follow the principles of heredity and evolution in biology. The fundamental principle of evolutionary algorithm is Darwinian natural selection in nature—'the survival of the fittest'. From the point of mathematics, these algorithms belong to search method rather than analytical method. Based on a set of random initial feasible solutions, the optimum solution will emerge eventually one generation after another. The measures to improve the quality of each generation are genetic operations such as reproduction (selection), crossover (recombination) and mutation.

It is well known that production planning plays an important role in mining engineering. The planning indicates the mining sequence and mining amount from each block at each year. This paper proposes a new approach to optimize production planning in underground mines with the help of evolutionary algorithms.

The new approach involves the applications of genetic algorithm and evolutionary programming. The approach is composed of two phases. Firstly, a general production planning is roughly outlined with the help of genetic algorithm. Then, the planning is modified by means of evolutionary programming according to the technical requirements of mining engineering. In other words, the purpose of the first phase is to determine the blocks to be mined in each year, while the second phase is to determine the ore amount to be mined in the blocks.

Evolutionary computing has come a long way, and some acknowledgement should be given to the pioneers of the method[2,3].

General planning by genetic algorithm

The purpose of this phase is to select a set of blocks for mining in each year, which will reach the goal to maximize the benefit meanwhile also meet the basic production requirements roughly such as production rate and mining sequence. The method in this phase is genetic algorithm.

Coding

Usually binary strings are used to represent the problem in genetic algorithms, which is equivalent to chromosome in biology to deliver genetic messages.

In order to represent the mining sequence, this paper uses binary strings with $y_i = n \cdot m$ symbols, where n represents total number of blocks in the underground mine, m stands for the year of the block to be mined, and i represents the i-th individual. For example, if $n = 3$ and $m = 4$, individual:

$y_i = 0010\ 0100\ 0101$

implies that its first block will be mined in the second year, its second block and the third block will be mined in the fourth year and the fifth year respectively.

The initial population

In order to carry out parallel searching with multiple points, there are always 50~100 individuals as a population in genetic algorithms. As the first generation, the initial population is usually generated by random combination of binary symbols, i.e. to determine randomly the year of each block to be mined in the initial stage.

However, the initial population must meet the requirements on ore amount and metal amount, i.e.:

$$\sum_{j=1}^{N} x_{ijt} = A_t \qquad i = 1\ 2 \ldots M \qquad t = 1\ 2 \ldots T \qquad [1]$$

$$\sum_{j=1}^{N} x_{ijt}\ q_j = Q_t \qquad i = 1\ 2 \ldots M \qquad t = 1\ 2 \ldots T \qquad [2]$$

where: x_{ijt}—mining amount of the j-th block in the i-th individual at year t. It is also the ore amount owned by the block in order to make a general decision for genetic algorithm.

M number of individuals in a population
T life of the mine
N number of blocks mined at year t
q_j average grade of block j.

In order to avoid the generation of unqualified individuals, the initial individuals can be assigned according to the natural sequence of blocks in orebody. In other words, the initial individuals are the blocks which locate at the upper level and close to main development

opening. Figure 1 shows an individual, where number in blocks represent the year the block is to be mined,

Fitness

Fitness is the driving force to promote evolution in genetic algorithm. It is also the object function to evaluate the quality of different individuals. In this paper, fitness is net present value of individual, i.e.:

$$P = \sum_{t=1}^{T} x(I_t - O_t)(1+r)^{-t} \qquad [3]$$

where P net present value of each individual (mining alternative)
I_t cash flow-in at year t
O_t cash flow-out at year t
r discount rate
T life of the mine.

Cash flow-in at year t I_t is calculated as follows:

$$I_t = \sum_{N} x_{ijt}\, q_j\, \alpha p \qquad [4]$$

where p specified price of metal
α recovery rate of metael
other symbols— the same as before.

Cash flow-out at year t is calculated as follows:

$$Q_t = H_t + T_t + V_t + D_t + K_t \qquad [5]$$

where H hoisting cost at year t, calculated according to the vertical coordinate z of blocks
T_t transportation cost at year t, calculated according to the horizontal location x,y of blocks
V_t ventilation cost at year t, calculated according to the spatial location x,y,z of blocks
D_t drainage cost at year t
K_t development and maintenance cost for openings at year t.

Although the calculation of fitness here is not precise enough, it is still satisfied with the comparison of quality for different individuals.

Reproduction

By means of reproduction in genetic algorithms, some individuals with excellent quality will be copied to the next generation while the same number of bad individuals will be deleted. In this way Darwinian Principle 'the survival of the fittest' is fulfilled.

Fitness-proportional selection is used as the reproduction method in this paper. The probability for individual to be copied into the next generation is determined by its fitness. Of course, the better the individual is, the larger the probability will be. However, a few bad individuals could be copied to the next generation so that the population will have more variety to promote evolution. The probability can be calculated as follows:

$$P_i = \frac{f_i}{\sum_{M} f_i}$$

where: P_i probability of individual i to be copied
f_i fitness of individual i
M number of individuals in population.

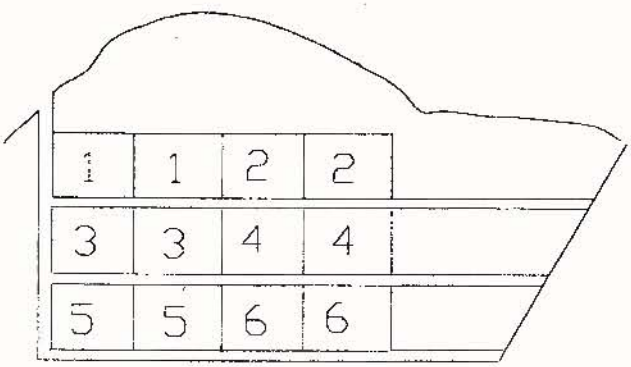

Figure 1. Generatincy an initial individual

Crossover

Crossover is the main operation to generate new individuals in genetic algorithm. New individuals are resulted from the exchanging of some symbols in old individuals, which is similar to the crossover in biology.

In genetic algorithm it is a random selection to choose individuals and location for crossover. However, multiple-point crossover is used in this paper since the symbolic strings are rather long. By means of multiple points the new chromosome can be spread along the length of the string. Table I illustrates the process of multiple-point crossover.

The new individuals generated by crossover may not satisfy with the technique constraints from the view of mining engineering, therefore all the new individuals must be checked according to constraint Equations [1] and [2]. The unqualified individuals after checking will be deleted, and new individuals will be regenerated by crossover from other parents.

Mutation

Mutation is another operation to generate new individuals in genetic algorithm. New individuals will emerge by random change of a symbol in some individuals, i.e. to change a symbol from 1 to 0 or from 0 to 1.

The new individuals generated by mutation might not satisfy with the technical requirements as in crossover. Therefore new individuals will be regenerated to replace the unqualified one.

Termination

Genetic algorithm is an iterative computation. It is necessary to determine some criteria to terminate the iterations. There are two criteria applied to terminate the algorithm in this paper:

- Set up maximum iterative times. As soon as the iterations reach the given maximum times, the algorithm will be forced to stop and output the optimum result
- Observe the variation of fitness. When the variations of the maximum fitness or average fitness become stationary, it can be regarded as converge and the algorithm will be terminated.

Table I
Multiple-point crossover

Crossover point					1				2				3			
Parent 1	0	1	0	0	**1**	1	0	1	**1**	0	0	1	**0**	1	0	1
Parent 2	1	0	0	1	**0**	0	1	0	**0**	1	0	1	**1**	0	1	0
Children 1	0	1	0	0	**0**	0	1	0	1	0	0	1	**1**	0	1	0
Children 2	1	0	0	1	**1**	1	0	1	**0**	1	0	1	**0**	1	0	1

Modification by evolutionary programming

The purpose of this phase is to modify the general planning after genetic algorithm with the help of evolutionary programming. The object of this phase is to determine the exact ore amount mined by each block at each year rather than only a year as in the previous phase. The fitness and constraints are similar to the previous phase.

Representation of the problem

In evolutionary programming, decimal digits are used to represent the problem rather than binary digits as in genetic algorithms[4]. The decision variable x is modified by a random variance σ.

In this paper the problem is described as follows:

$$(X,\sigma)=\left((x_1,x_2,\cdots,x_n),(\sigma_1,\sigma_2,\cdots,\sigma_n)\right)$$

$$\text{where: } x_i' = x_i + \sqrt{\sigma_i}N_i(0,1) \qquad [6]$$

$$\sigma_i' = \sigma_i + \sqrt{\sigma_i}N_i(0,1) \qquad [7]$$

where: (x_i,σ_i) the i-th individual of parent generation
(x_i',σ_i') the i-th new individual of children generation
x_i amount mined at block i
σ_i variance of block i
$N_i(0,1)$ random number generated for individual i, which obeys standard normal distribution function
n number of blocks in individual.

Hence, the new ore amount x_i' mined by block i is determined by the original amount x_i with a random disturbance.

Generation of the initial population

Similar to genetic algorithm, the initial individuals in evolutionary programming result also from random generation. Their decimal digits and variances are selected by random process.

However, the initial individuals are also required to meet the constraints on ore demand and metal demand as in Equations [1] and [2]. In order to meet these requirements, the following tricks are used:

- Starting from the first year, the modification of production planning by evolutionary programming is carried out one year after another. In other words, the purpose of evolutionary programming is for the optimization in each year, which is based on the global optimal result by genetic algorithm
- The blocks involved in modification for each year are limited to the optimal result by genetic algorithm. However, when it is necessary the blocks involved can be extended to the global optimal blocks either in the previous year or the next year
- Variance σ_i can be determined according to the deviation from the constraints [1] and [2], i.e.:

$$\sigma_i = \frac{A_t - \sum_n x_i}{n}$$

where: n number of blocks involved in each year by evolutionary programming; other symbols—the same as above.

Calculation of fitness

In this paper fitness of evolutionary programming is benefit R of the mine in the involving year t, i.e.:

$$R = I_t - O_t$$

where: I_t cash-flow in, calculated according to Equation [4]
O_t cash-flow out, calculated according to Equation [5].

Mutation

In evolutionary programming, mutation is the unique measure to generate new individuals. New individuals are obtained by random disturbance from old individuals as in Equations [6] and [7].

However, it is necessary to check σ_i if it is larger than 0 since there is $\sqrt{\sigma_i}$ in equations. If $\sigma_i < 0$, then let σ_i to be ξ_0, which is a small number but larger then 0.

Selection

In evolutionary programming there is no crossover as in genetic algorithm. After mutation there will be genetic operation called selection, which is similar to reproduction in genetic algorithm. By means of selection, μ individuals are selected both from the old population and new population (total 2μ individuals).

In this paper q-competition selection is used[5]. In order to determine individual i if it can be copied into the next generation, q individuals are selected randomly both from the old generation and the new one as a testing group, then individual i is compared with all the individuals in the testing group respectively and record the times when the fitness of individual i is better than the individual in testing group, it is the score $\overline{w_i}$ of individual i, i.e.:

$$\overline{w_i} = \sum_{j=1}^{q} \begin{cases} 1 & \text{if } f_i \text{ is better than } f_j \\ 0 & \text{others} \end{cases}$$

where: f_i fitness of individual i
f_j fitness of the j-th individual in testing group.

Finally, the individuals with higher scores are selected as the seed and copied into the next generation.

Termination

Evolutionary programming is also an iterative algorithm and there should be a termination criterion to stop the computation as in genetic algorithm.

In this paper the termination criteria in evolutionary programming are similar to that in genetic algorithm, i.e. according to the times of iterations and the variation of fitness.

Example

As an example in this paper, an underground mine with 5 levels and 80 blocks are illustrated for the optimization of production planning by evolutionary algorithms. The life of the mine is 20 years. The original distribution of blocks is shown in Figure 2.

By means of genetic algorithm, the general planning as the first phase is shown in Figure 3, where the digits represent the year for the blocks to be mined. The computation parameters of genetic algorithm are as follows:

Number of individuals in a population M	100
Reproduction rate P_r	0.2
Crossover rate P_c	0.6
Mutation rate P_m	0.05
Maximum number of iterations T	80

Based on the result after genetic algorithm, modification

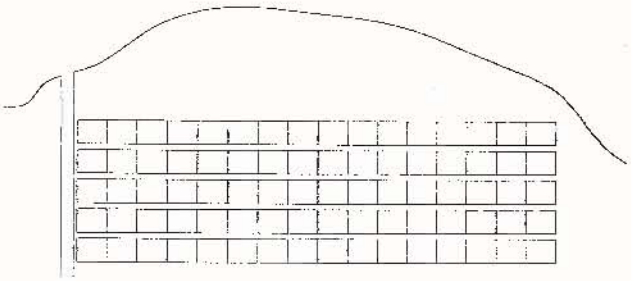

Figure 2. Distribution of blocks

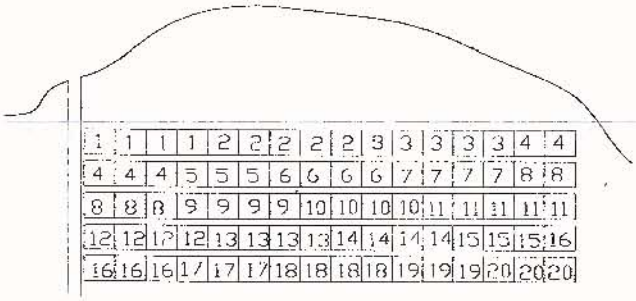

Figure 3. Result by genetic algorithm

for production planning is carried out by evolutionary programming. Figure 4 illustrates the result of modification. The parameters of evolutionary programming are as follows:

Number of individuals in a population μ	100
Number of testing group q	0.9
Maximum number of iterative times T	100

Conclusions

Evolution algorithms are a new kind of optimization techniques, where there is no need to write down a rigorous comprehensive mathematical model as a primary requirement. The algorithm can obtain the final optimal solution by means of self-adaptive searching. It is particularly suitable to solve the complicated and complex problems in mining engineering.

This paper provides a new approach to optimize the production planning in underground mines with the help of evolutionary algorithms and computer techniques. The new approach consists of two phases. Firstly, genetic algorithm is used to determine the year for each block to be mined. Then, evolutionary programming is employed to modify the mined amount for each block. In other words, the first phase is to reach the global optimization while the second

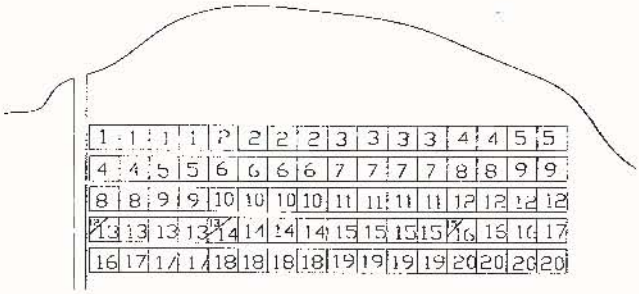

Figure 4. Result by evolutionary programming

phase is to obtain the partial optimization for each year.

The new approach is different from the traditional method such as linear programming or integer programming[6]. The advantage of the new approach is that it is not necessary to set up a rigorous mathematical model such as object function and constraint equations, and there is also no need to use complex simplex algorithms and branch and bound algorithm.

The application indicates that the new approach is successful and reasonable. It provides a new tool for the production planning in underground mines.

References

1. BACK, T. *Evolutionary Algorithms in Theory and Practice*: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York. 1996.
2. DENBY, B. and SCHOFIELD, D. Open-pit design and scheduling by use of genetic algorithms. *Transactions of the Institution of Mining and Metallurgy, Section A: Mining Industry*, 1994, vol. 103, pp. A21–A26.
3. JUNG, J.H., LEE, C.H., LEE, I.-B. Genetic algorithm for scheduling of multi-product batch processes. *Computers & Chemical Engineering*. October 1998, vol. 22, issue 11, pp. 1725–1730.
4. SCHWEFEL, H.-P. *Evolution and Optimum Seeking*, John Wiley & Sans, Inc. New York. 1995.
5. YUN, Q.X. *Evolutionary algorithms*. Metallurgical Publication Press. Beijing. 2000 (in Chinese).
6. YUN, Q.X., LIU, J.X., CHEN, Y.F., and HUANG, G.X. Optimization of planning and design in underground mines. *Proceedings of the 22nd Symposium on the Applications of Computer and Operations Research in the Mineral Industries* (APCOM). 1992. Germany. pp. 255–262.