Improved Genetic Algorithm for Integrated Steelmaking Optimum Charge Plan

Yun-Can Xue[1,3], Xin Wang[2], Shao-Yuan Li[3]

[1]*College of Computer & Information Engineering, Hohai University, Changzhou, China*
[2]*Institute of Information & Control Technology, Center of Electrical & Electronic Technology, Shanghai Jiao Tong University, Shanghai, China*
[3]*Institute of Automation, Shanghai Jiao Tong University, Shanghai, China*

Abstract: The shortcoming of the standard genetic algorithm is analysed. An improved genetic algorithm with modified mutation operator and adaptive probabilities of crossover and mutation is proposed. Simulation experiments have been carried and the results show that the modifications are very effective. In this paper, an optimum charge plan for steelmaking continuous casting production scheduling is also studied. The charge plan model is established. The modified genetic algorithm is used to solve the optimum charge plan problem. The computation with practical data shows that the model and the modified genetic algorithm are very effective.

Keywords: steel manufacture, modelling, scheduling algorithms, genetic algorithms, operators, optimisation.

1. INTRODUCTION[1]

Iron and steel industrial is an essential and sizable sector for industrialized economies. Since it is capital and energy extensive, companies have been putting consistent emphasis on technology advances in the production to increase productivity and to save energy. The modern integrated process of steelmaking continuous casting and hot rolling directly connects the steelmaking furnace, the continuous caster and the hot rolling mill with hot metal flow and makes a synchronized production. However, it also brings new changes for production planning and scheduling. For steel making process, the main work is to arrange the charge plan and cast plan. The basic unit of steelmaking is the charge. To make the charge plan, the following conditions are needed:

1) steel grades must be the same
2) steel thicknesses of slabs in the same charge must be equal,
3) the slab width must be the same,
4) the consignment date must be near.
5) the total weight in each furnace must be greater than 90% furnace capacity and less than the 100% furnace capacity.

The mathematical models of the optimum charge plan are as follows (Tang et al. 1996):

$$\min .Z = \sum_{i=1}^{N}\sum_{j=1}^{N}(C_{ij}^1 + C_{ij}^2 + C_{ij}^3)X_{ij} + \sum_{j=1}^{N} p_j * Y_j + \sum_{j=1}^{N}(1 - X_{ij})h_j \qquad (1)$$

$$\text{s.t.} \quad \sum_{j=1}^{N} X_{ij} \le 1, i = 1,..., N \qquad (2)$$

$$\sum_{j=1}^{N} X_{ij} = P \qquad (3)$$

$$\sum_{j=1}^{N} g_i * X_{ij} + Y_j = T * X_{jj}, j = 1,..., N \qquad (4)$$

$$X_{ij} \le X_{jj}, i = 1,..., N, j = 1,..., N \qquad (5)$$

$$Y_j \ge 0, j = 1,..., N \qquad (6)$$

$$X_{ij} \in \{0,1\}, i = 1,..., N, j = 1,..., N \qquad (7)$$

Where:

P—charge number.

[1]This work was supported by China 863 High Technology Item (no. 2002AA412010), the National Natural Science Foundation of China (70201004) and the Key Technology and Development Program of Shanghai Science and Technology Department (04DZ11008)

N--the slab number to be arranged.

T—the furnace capacity.

p_j—annexed cost coefficient of residual armor plate of the jth charge number.

g_j--the weight of the jth slab.

h_j—annexed cost coefficient of the jth slab not be chosen.

Y_j —The open order of the the jth charge.

W—weight of the i*th slab*.

C_{ij}^2 --annexed width cost coefficient of slab i combined to slab j and:

$$C_{ij}^1 = \begin{cases} 0 & \text{Steelgrades of slabs } i \text{ \& } j \text{ are equal} \\ F_1(ST_i - ST_j) & \text{slab } i \text{ \& } j \text{ belong to the same armor plate serial and the requirement of } slab\ i \text{ is lower than } slab\ j \\ +\infty & \text{slab } i \text{ \& } j \text{ belong to the same armor plate serial and the requirement of } slab \text{ i is higher than } slab\ j \\ +\infty & \text{slab } i \text{ \& } j \text{ donot belong to the same steelgrade serial} \end{cases} \quad (8)$$

$$C_{ij}^2 = \begin{cases} 0 & W_i = W_j \\ F_2 * |W_i - W_j| & 0 < |W_i - W_j| \le E \\ \infty & |W_i - W_j| > E \end{cases} \quad (9)$$

C_{ij}^3 --annexed date cost coefficient of slab i combined to slab j and:

$$C_{ij}^3 = \begin{cases} F_3(d_i - d_j) & d_i - d_j \ge 0 \\ F_4(d_j - d_i) & d_j - d_i < 0 \end{cases} \quad (10)$$

3. GENETIC ALGORITHM FOR OPTIMUM CHARGE MODEL

3.1. Simple Genetic algorithm and its shortcoming

GA is a general methodology for searching a solution space in a manner analogous to the natural selection procedure in biological evolution (Holland, 1975). GA differs from many traditional optimization algorithms in that the latter usually suffer from myopia for highly complex search spaces (Miller, 1993 and Salomon 1998). The prominent characteristic of GA is that it can test and manipulate a set of possible solutions simultaneously, which assures GA to find the optimal solution, which cannot be found by “hill-climbing” search algorithms or “gradient descent” techniques.

Although GA has been successfully used in many areas, such as machine learning (Englander, 1985), neural network (Eshelman,1991) and TSP (Grefenstette,1985), etc., there remain problems needed to develop in GA, i. e. premature convergence (Schraudolph,1992, Potts,1994, Back,1991 and Eshelman,1991).

Prior researchers have made efforts to prevent premature convergence including improving selection strategy (Syswerda,1989,Davidor, 1989), crossover model (Fogarty,1989,Srinivas,1994) and probabilities of crossover and mutation (Syswerda,1989,Goldberg,1990 and DeJong ,1985).

In the usual version, mutation operator has ability to exploit the critical alleles. So researchers seldom suspected the ability of the traditional mutational operator to prevent premature convergence and they have been ignoring to improve the traditional mutation operator. As we all know, new chromosomes can be generated after crossover, but no novel genes can be yielded because the operation of crossover is just to exchange parts of genes between parents. Selection strategies can bring neither new chromosomes nor new genes into population. In the stage of selection, GA only selects the higher fitness chromosomes from the contemporary population to reproduce. So GA cannot generate new genes for some loca after reproduction. On the contrary, critical alleles in some loca will disappear with the death of “bad” individuals because of selection. Therefore the exploitation of critical alleles depends on the mutation operator. The traditional mutation operator performs *NOT* operation. This kind of genetic operator, on the one hand, is beneficial to find the critical alleles when premature convergence appears, but on the other hand, it may hurt the critical alleles while mutation acting on the critical alleles.

3.2. An improved mutation operator

As we all know, invalid genes occupy some loca when GA converges prematurely. To prevent premature convergence, it’s important to maintain the diversity of genes in the same locus rather than the diversity of individuals in the population. Since we cannot identify which kind of genes is critical in a certain locus, we had better enable the alleles to exist in the same locus during the period of mutation.

Here, we present a new mutation operator, which is made up of two boolean operators: $XOR / \overline{XOR}$. The expression of the boolean operators is as

$$XOR: \begin{cases} a \otimes b = 0, & if \quad a = b \\ a \otimes b = 1, & if \quad a \ne b \end{cases}$$

$$\overline{XOR}: \begin{cases} a \circ b = 1, & if \quad a = b \\ a \circ b = 0, & if \quad a \ne b \end{cases}$$

This differs from the traditional one in that the latter is made up of one boolean operator: *NOT*. Obviously mutation with new genetic operator needs parents to provide two genes. According to (8), the result of mutation is that the mutated genes in the same locus of two offspring are in the state of compensation. So provided that there is a pair of genes mutated in the locus, there will be at least one critical allele coming into being in the same locus after mutation. The probability of the loss of critical alleles caused by the improved mutation operator can reduce to zero. As a result, the new mutation operator can prevent premature convergence to a high degree.

Here is an example that the genes in the 4th and 7th locus undergo mutation respectively:

Parents	*Offspring*	*operator*
0111010	0110011	*XOR*
1101011	1101010	$\overline{XOR}$

Before mutation, there are two different genes in the 4th locus and genes in the 7th locus are the same while they are mutually exclusive in their own locus after mutation.

3.3. Adaptive probabilities of crossover and mutation

The significance of crossover probability and mutation probability in controlling GA performance has long been acknowledged in GA research (Srinivas, 1994, 1989, Goldberg, 1990). Several studies both empirical and theoretical have been devoted to identify optimal parameter settings for GAs. The bigger the p_c , the quicker the new solutions are introduced into the population. As p_c increases, however, solutions can be disrupted faster than selection can exploit them. Typical values of p_c are in the ranges 0.5-1.0. Mutation is only a secondary operator to restore genetic material. Nevertheless the choice p_m is critical to GA performance and has been emphasized in DeJong's inceptive work (DeJong, 1985). Large p_m transforms the GA into a purely random search algorithm, while some mutation is required to prevent the premature convergence of the GA to suboptimal solutions. Typical values of p_m are in the range 0.05-0.2.

In order to improve the performance of the GAs in optimizing multimodal functions, a lot of works have been done (Srinivas, 1994, Goldberg, 1990, DeJong, 1985). DeJong introduced the ideas of "overlapping populations and 'crowding' in his work (DeJong, 1985). Goldberg proposed a Boltmann tournament selection scheme for forming and sizing stable sub-populations (Goldberg, 1990). Srinivas has proposed the adaptive probabilities of crossover and mutation in Gas (Srinivas, 1994).

The idea of adaptive operators to improve GA performance has been employed earlier (Srinivas, 1994, Schaffer, 1987). Among them, Sirnivas's work is very effective. In his work, the expressions for p_c and p_m are given as:

$$p_c = k_1(f_{\max} - f')/(f_{\max} - f_{avg}),\ f' \geq f_{avg}, \tag{11}$$

$$p_c = k_3,\ f' < f_{avg} \tag{12}$$

$$p_c = k_2(f_{\max} - f)/(f_{\max} - f_{avg}),\quad f \geq f_{avg}, \tag{13}$$

$$p_m = k_4, f < f_{avg} \tag{14}$$

where:

$$k_1, k_2\ k_3\ k_4 \leq 1.0 \text{ and } \quad f' = \max(f_1, f_2) \tag{15}$$

From (11)-(13), it can be seen that the more nearly the individual fitness approaches the biggest fitness, the smaller the p_c and p_m . When the individual fitness equal to the biggest fitness, the values of p_c and p_m take zeros. Also in the later evolution period, big changes are not suitable to the individual near the optimum to prevent the better performance of the individuals to be demolished, this will lead to no changes are made to the better individual. However, the better individual maybe is not the global optimum. This will increase the probability of the evolution process to stop at the local optimum.

To overcome the above stated problem, based on the discussion in section 3.2, here we proposed the following adaptive p_c and p_m formula:

$$p_c = p_{c1} - (p_{c1} - p_{c2})(f - f_{avg})/(f_{\max} - f_{avg}),$$

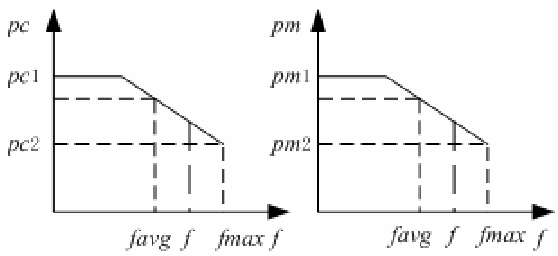
$$f \geq f_{avg}, \tag{16}$$

$$p_c = p_{c1},\quad f < f_{avg} \tag{17}$$

$$p_m = p_{m1} - (p_{m1} - p_{m2})(f - f_{avg})/(f_{\max} - f_{avg})$$

$$f \geq f_{avg}, \tag{18}$$

$$p_m = p_{m1},\ f < f_{avg} \tag{19}$$



a) crossover operator b) mutation operator

Fig.1 Adaptive crossover and mutation operators

where:

$$f = (f_1 + f_2)/2 \tag{20}$$

In order to test the performance of improved GA with adaptive probabilities of crossover and mutation and modified operator (in brief IGA), the following mathematical functions (Srinivas, 1994) are introduced to test the IGA:

$$f_1(x) = 4 + x_1^2 + x_2^2 - 2\cos(2\pi x_1) - 2\cos(2\pi x_2)$$

$$-5 \le x \le 5 \qquad (21)$$

$$f_2(x) = 0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}$$

$$-5 \le x \le 5$$

a[2][25]=
{{-32,-16,0,16,32,-32,-16,0,16,32,-32,-16,0,16,32,-32,-16,0,16,32,-32,-16,0,16,32},
{-32,-32,-32,-32,-32,-16,-16,-16,-16,-16,16,16,16,16,16,32,32,32,32,32,0,0,0,0,0}}
(22)

$$f_3(\mathbf{x}) = (x_1^2 + x_2^2)^{0.25}\left[\sin^2(50((x_1^2 + x_2^2))^{0.1}) + 1.0\right]$$

$$-5 \le x \le 5 \qquad (23)$$

The population sizes in both standard genetic algorithm (in brief, SGA) and IGA are 30, and the length of encoded strings is 10 for each variable. The probability of crossover p_c ranges from 0.6-0.9, and the probability of mutation p_m ranges from 0.05-0.2.

Simulation results are show in Fig.2. Fig.2 (a),(b),(c) represent the search processes of function f_1, f_2, f_3 respectively. Where the dashed lines represent the search process of SGA and the solid lines represent those of the IGA. From the figures, it can be seen that the IGA searches much fast than the SGA.

3.4. Improved genetic algorithm for optimum charge plan

Based on the above discussion, the IGA is used to solve the optimum charge plan problem. The charge model solution has following properties:

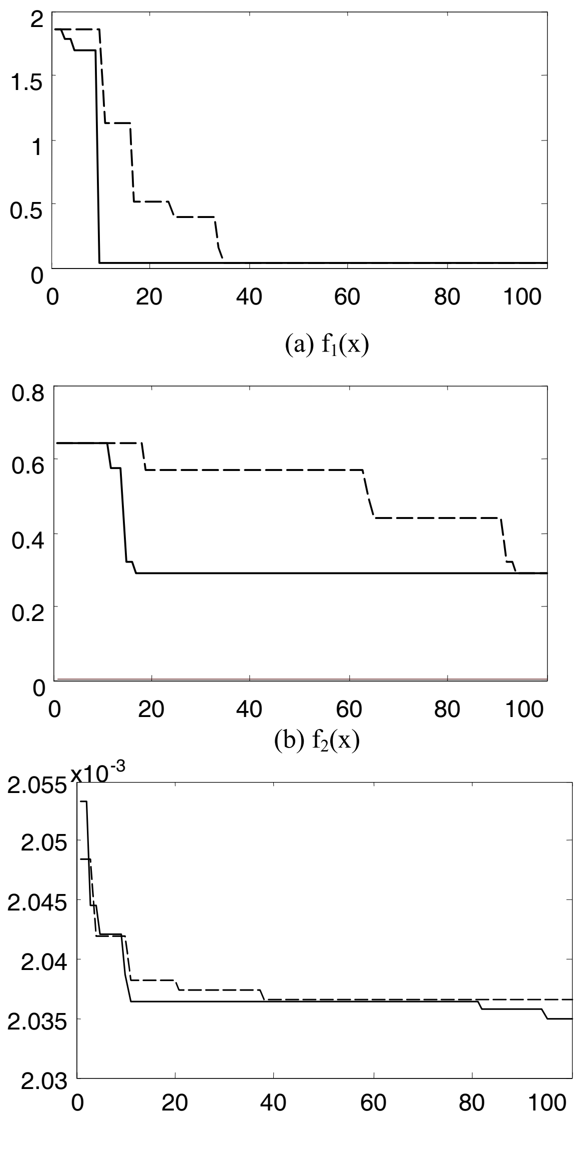
(a) Property of the charge model solution

The configuration of the solution is with the 2-dimensions matrix form. The row represents slab number, the column represent charge number.

Only one element in each row is 1, the others are 0.

(b) Construction of the chromosome

Assume $X = (a_1, a_2, \cdots, a_N)$, the element index in X represents the charge number and the element represents the charge number, and $a_i \in \{1,2,\cdots LA\}$ where $i = 1,2,\cdots,N$.



(a) $f_1(x)$

(b) $f_2(x)$

(c) $f_3(x)$

Fig.2 Search process of SGA and ISGA

(c) Fitness function

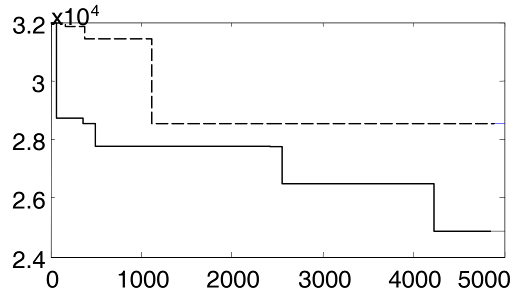


Fig.3 SGA and IGA search process of charge plan

4. APPLICATION EXAMPLE

According to (1), let $f = C_{\max} - z$, this changes the minimize problem into the maximum problem and $C_{\max}$ is a large constant.

Now take the practical data in an Iron and steel plant as an example. The basic model parameters are listed in Table 1.There are 15 slabs to be arranged into 4 charge number. According to the charge model and the improved genetic algorithm, the results are listed in Table 2.Fig. 3 shows the response process of SGA and IGA. From the figure, it is obvious that the IGA response much fast than that of the SGA.

Table 1. Basic model parameters and the computation results

Contract number	Steelgrade serial	Steelgrade	Width	Consignment date	Weight
1	13	DT5427A1	1123	4	31.7
2	12	DT5427A2	1200	5	29.7
3	14	DT5427A4	950	6	28.7
4	12	DT5427A2	1178	7	29.5
5	24	AP1055E4	1150	4	29.7
6	21	AP1055E1	1135	5	29.5
7	23	AP1055E3	1168	6	29.6
8	21	AP1055E1	1046	7	28.5
9	11	DT5427A1	1200	5	26.8
10	12	DT5427A2	1250	6	35.3
11	14	DT5427A4	1250	7	29.5
12	12	DT5427A2	950	8	27.5
13	21	AP1055E1	1213	5	31.5
14	23	AP1055E3	1046	7	32.5
15	21	AP1055E1	1300	8	32.1
T=100 E=100 F1=4 F2=0.1 F3=20 F4=15 P=20 H=100					

Table 2. Basic model parameters and the computation results

Contract number	Steelgrade serial	Steelgrade	Width	Consignment date	Weight	Charge number
1	13	DT5427A1	1123	4	31.7	1
2	12	DT5427A2	1200	5	29.7	1
3	14	DT5427A4	950	6	28.7	0
4	12	DT5427A2	1178	7	29.5	1
5	24	AP1055E4	1150	4	29.7	3
6	21	AP1055E1	1135	5	29.5	4
7	23	AP1055E3	1168	6	29.6	3
8	21	AP1055E1	1046	7	28.5	4
9	11	DT5427A1	1200	5	26.8	2
10	12	DT5427A2	1250	6	35.3	2
11	14	DT5427A4	1250	7	29.5	2
12	12	DT5427A2	950	8	27.5	0
13	21	AP1055E1	1213	5	31.5	3
14	23	AP1053E3	1046	7	32.5	4
15	21	AP1055E1	1300	8	32.1	0
T=100 E=100 F1=4 F2=0.1 F3=20 F4=15 P=20 H=100						

5. CONCLUSION

Based on the charge production process, a charge model is presented by considering the effects of steelgrade, slab width and consignment date. This model is very practical and is easy to be used. To solve the optimum charge plan, an improved genetic algorithm with improved mutation operator is proposed. Simulation results with practical iron and steel plant data show that the model and computation method is very useful and effective and can give good charge plan.

REFERENCES

Back T. and F. Hoffmeister (1991). Extended selection mechanisms in genetic algorithms. In: *Proc. 4th Int. Conf. Genetic Algorithms*, Univ of California, San Diego, CA, pp.14-21

Davidor, Y. (1989). Analogous crossover. In: *Proc. 3rd Int. Conf. Genetic Algorithms*, George Mason Univ., Arlington, VA, 1989, pp. 98-103.

DeJong, K. A. (1985). Genetic algorithms: A 10 years perspective. In: *Proceedings of an International conference of genetic algorithms and their applications*, Pittsburgh, pp.169-177.

Englander, (1985), Machine learning of Visual Recognition Using Genetic Algorithms. In: *Proc. Int. Conf. Genetic Algorithms and Their Application*, pp.160-168.

Eshelman L. J. and J. D. Shaffer. (1991), Preventing premature convergence in Genetic algorithms by preventing incest. In: *Proc. 4th Int. Conf. Genetic Algorithms*, pp.115-122.

FoGArty, T. C. (1989), Varying the probability of mutation in the genetic algorithm In: *Proc. 3rd Int. Conf. Genetic Algorithms,* pp.104-109.

Goldberg, D. E. (1990), A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. In: Complex system. Vol. 4, pp.445-460.

Grefenstette, J. J. , R. R. Gopal and D. Van Gucht. (1985), Genetic Algorithms for the Traveling Salesman Problem. In: *Proc. Int. Conf. Genetic Algorithms and Their Applications*, pp. 160-168.

Holland, J. H. (1975) . *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. MichiGAn Press.

Miller, J. A., W. D. Potter and R. V. Gandham (1993). An evaluation of local improvement operators for genetic algorithms. In: *IEEE Trans. Syst., Man, Cybern*., vol. SMC-23, no.5, pp. 1340-1350.

Potts, J. C. T. D. Giddens and S. B. Yadev (1994). The development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial Selection. In: *IEEE Trans. Syst., Man, Cybern.*, vol.SMC-24, no.1, pp. 73-85.

Richard, K. B. (1991). Evolving Networks: Using the Genetic Algorithms with Connectionist Learning. In: *Artificial Life 2*, Reading, MA: Addison-Wesley, pp. 511-547.

Salomon, R. (1998). Evolutionary algorithms and gradient search: similarities and differences. In: *IEEE Trans*. Evolutionary Computation, vol.2, no.2, pp. 45-55.

Schaffer J. D. and A. Morishma (1987). An adaptive crossover mechanism for genetic algorithms. In Proceedings of the second international conference on genetic algorithm,pp.36-40.

Schraudolph N. N. and R. K. Belew (1992). Dynamic Parameter encoding for genetic algorithms. In: *Machine learning*, vol.9, no.1, pp. 9-21.

Srinivas, M. and L. M. Patnaik (1994). Adaptive probabilities of crossover and Mutation in genetic Algorithms. In: *IEEE Trans. Syst. Man, and Cybern*., vol. SMC-24, no.4, pp.656-666.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In: *Proc. 3rd Int. Conf. Genetic Algorithms*, pp. 2-9.

Tang L. , Z. Yang and M. Wang (1996), Model and Algorithm of Furnace Charge Plan for Steelmaking-Continuous-Casting Production Scheduling. In: Journal of Northeastern University (Natural Science),vol.17, no.4, pp.440-445.