Optimization of Strip Hot Rolling Scheduling using a Genetic Algorithm

Carlos A. Hernández Carreón[1], Héctor J. Fraire Huacuja[2], Karla Espriella Fernandez[1], Guadalupe Castilla Valdez[2], Juana E. Mancilla Tolama[1]

[1]IPN, E.S.I.M.E. Azcapotzalco
Av. de las Granjas No. 682 Col. Santa Catarina
02250-Azcapotzalco, México, D. F.
[2] Instituto Tecnológico de Ciudad Madero.
1o. de Mayo y Sor Juana I. de la Cruz S/N.
89440-Cd. Madero, Tamaulipas, México
cahc05@yahoo.com.mx, hfraire@prodigy.net.mx,
karla_rouge@yahoo.com, gpe_cas@yahoo.com.mx,
emtolama@hotmail.com

Abstract. In this paper the problem of minimizing the hot rolling time of a steel strip is approached. A solution method based on genetic algorithm (GA) is proposed. The proposed algorithm uses SBX crossover, uniform mutation and initial population with feasible and infeasible individuals. To validate the approach a six-stand rolling mill is modeled as an optimization constrained problem and a set of problem instances were built using industrial data. The global rolling time obtained with the proposed GA is 0.051% better than the industrial time and obtains the best solution only in 23.62 CPU seconds. A relevant feature of this proposal is that the rolling schedule generated with the GA diminishes the equipment damage risks because it produces softer reductions than the rolling schedule proposed by the manufacturer. Currently we are developing new solution methods using different metaheuristics.

1. Introduction

Steel hot rolling is one of the most important metalworking processes in comparison with any other deformation process, aimed to manufacture products of relatively large dimensions (sheets, strips, plates, foils, etc.), at high speeds [1]. A rolling mill reduces the thickness steel slab in cages with pairs of work and support rolls as we can see in Fig. 1. Due to high operational costs of a rolling mill is not acceptable to setup the rolling schedule in an empirical way.

The hot rolling scheduling problem consist in determining the reductions for every rolling pass to obtain the final thickness, such that the rolling power should be lower than the nominal motor power. The hot rolling process has been approached in several research works, where different heuristics were applied. Some approaches consist in optimizing the rolling schedule to get strips of good quality, others in reducing the damage on the rollers and there are jobs where productivity of the system is maximized. In this work a Genetic Algorithm is applied to get a hot rolling schedule, for a six stands rolling mill. The GA was tested using industrial data.

© G. Sidorov, B. Cruz, M. Martínez, S. Torres. (Eds.) Received 23/03/08
Advances in Computer Science and Engineering. Accepted 26/04/08
Research in Computing Science 34, 2008, pp. 69-78 Final version 03/05/08

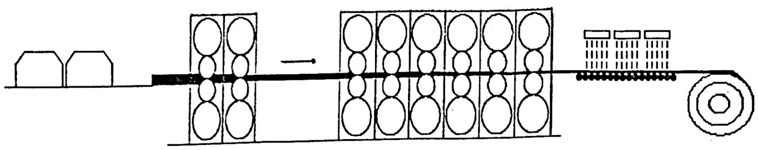

Fig. 1. Hot rolling process

2. Related Work

The application of metaheuristic algorithms to the hot rolling process has been the subject of several research works. Nolle & Armstrong [2] proposed the optimization of the surface quality of a steel slab in a 7-stand hot rolling mill using simulated annealing (SA) and genetic algorithms (GA). In this work the SA shows a better performance than GA. Oduguwa & Tiwari propose a general methodology to solve problems of sequential processes using GA [3]. The proposal consists in a binary representation of the full set of parameters as a sub-set of strings. An application to multi-pass hot rolling was given, using a multi-objective model. The goal was to maximize the system productivity, optimizing the roll force.

Chakraborti [4] applied GA to the problem of minimizing hot rolling time in a reversing mill stand, determining the optimum number of odd passes. In this work the efficiency of GA to calculate a hot rolling schedule, with respect to traditional methods, was demonstrated. Another contribution of Chakraborti [5] was the study of surface profiles of slab rolled. In this case, two objective functions were applied to evaluate the wearing and deflection rolls as the main factors of the variation of the thickness during rolling process. The GA produces good quality solutions with respect to the solutions corresponding to the industrial data [6]. Another approaches to determine hot rolling schedules have been applied as neural networks [7, 8], fuzzy logic [9, 10], and finite element methods [11, 12]. Currently one of the most successfully approaches to solve the hot rolling scheduling problem is the genetic algorithm.

Hernández apply a procedure to calculate the optimal pass reductions in two phases [13]. In the first one a non linear optimization function was applied to evaluate the computational cost of the problem solution. Then a GA was applied to compare two point and simulated binary crossover operators.

Unlike the traditional approaches in this work the steel chemical composition of the strip is incorporated, as a problem parameter. This will allow the automatically setup of the hot mill for different steels. A solution method based on genetic algorithm is proposed, which uses SBX crossover, uniform mutation and initial population with feasible and infeasible individuals.

3. Problem Description

3.1. Instance Description

Industrial data was obtained from the Hylsa Monterrey Company and from the software HSMM of INTEG Process Group [13]. Table 1 shows the different chemical compositions of the steels considered in this work. The first column contains the steel identifier. From column two to six are contained the weight percent of carbon, manganese, silicon, niobium, titanium and vanadium.

From the industrial data 17 instances were defined, which can be consulted in [14]. Each instance defines the parameters of a different rolling problem. A rolling problem consists in determining the intermediate reductions needed to roll the slab steel and to obtain the final thickness in a 6-stand roll mill. Table 2 shows the parameters included in an instance: the data source, the number rolling stands (n), the instance name, the initial thickness (h_0), the final thickness (h_f), initial width (w), chemical composition (%C, %Mn, %Si, %Mo, %Nb, %Ti, %V), and for each rolling stand the roll diameter (D_i), the roll speed (v_i), the temperature (T_i), the grain size (d_{0i}), the motor power (P_i) and the inter-stand distance (l_i) are indicated.

Table 1. Chemical composition of steel

Steels Id.	% C	% Mn	% Si	% Nb	% Ti	% V
1	0.0450	0.450	0.069	0.0056	0.002	0.080
2	0.0380	0.300	0.009	0.0050	0.002	0.002
3	0.0820	0.480	0.045	0.0360	0.002	0.002
4	0.0710	0.758	0.014	0.0230	0.013	0.003
5	0.0028	0.170	0.009	0.0350	0.035	0.005
6	0.0530	0.784	0.010	0.0260	0.000	0.000

Table 2. Hot rolling scheduling problem instance

Industrial Data: Hylsa		$n = 6$		Name: hyl001.txt	
h_0 =48 mm		h_f = 3.8 mm		w=991mm	
%C = 0.053, %Mn = 0.784, %Si = 0.017, %Mo = 0, %Ti = 0, %Nb = 0, %V = 0					
Roll pass	1	2	3	4	5	6
D_i (mm)	752	764	758	492	456	474
v_i (m/s)	0.81	1.43	2.21	3.38	4.57	5.54
T_i (°C)	1010	987.64	964.25	942.7	927.32	908.27
d_{0i} (μm)	400	100	80	60	40	20
P_i (kW)	7000	7000	7000	7000	7000	7000
l_i(m)	3.5	3.5	3.5	3.5	3.5	

3.2 Optimization Problem Formulation

Given an instance of the hot rolling scheduling problem, the goal is to determine the intermediate thicknesses $h_1,\ldots, h_{n-1}$ to minimize the total rolling time given by:

$$t = \sum_{i=1}^{n} t_i \tag{1}$$

The process time in each stand is accumulated, to calculate the total rolling time. The rolling time in each stand is calculated adding the contact time between the roll and the steel, and the time elapsed for the slab from one stand to another. The rolling time in a given stand i can be calculated as follows:

$$t_i = \frac{\sqrt{\Delta h_i \cdot R_i} + l_i}{v_i} \tag{2}$$

where:

R_i : roll radius in stand i.
l_i : inter-stand distance
v_i : roll speed in stand i.
Δh_i: $h_{i-1}-h_i$

The problem includes the following two constraints:

1. In every rolling stand a reduction is applied until the final thickness is obtained. Each intermediate thickness should be lower than the previous one:
$$h_0 > h_1 > h_2 > h_3 > h_4 > h_5 > h_f \tag{3}$$
2. To get the reduction in a rolling stand, a rolling power should be applied which be lower than the motor power:
$$W_i < P_i \qquad \text{for } i = 1 \ldots 6 \tag{4}$$

To calculate the rolling power W_i is required to calculate the flow stress. In this work we use the Hernández model to predict the flow stress of any kind of steel, independently of the chemical composition [15-18]. The flow stress parameters include the temperature, the strain, strain rate, grain size and the chemical composition of the rolling steel.

$$\sigma = \sigma\,(T, \varepsilon, \varepsilon, d_{0i}, \%C, \%Mn, \%Si, \%Mo, \%V, \%Ti, \%Nb) \tag{5}$$

In hot rolling, is better to calculate the deformation in terms of the bite angle α, so that we can calculate the resistance to deformation [19, 20] as follows:

$$\bar{k}_i = \frac{1}{\alpha_i}\int_0^{\alpha} \sigma_i d\alpha_i \tag{6}$$

Then the rolling forces to deform the steel can be calculated. The roll-separating force F can be calculated using different mathematical models, like rolling theories of Sims [21], Cook & McCrum [22] and Alexander & Ford [23], in this work is used the Alexander & Ford model

4. Genetic Algorithm Proposal

In this section the genetic algorithm proposed to solve the hot rolling scheduling problem is described. The GA determines the intermediate reductions required to obtain the final steel thickness in order to minimize the total rolling time. The GA uses SBX crossover [24], uniform mutation [25] and elitism. The initial population includes feasible and infeasible individuals.

4.1. Individual Representation Structure

Each individual is represented as a thicknesses vector $X = [h_0, h_1, h_2, h_3, h_4, h_5, h_f]$, where h_0 and h_f are the initial and final thicknesses of the strip and $h_1, h_2,..., h_5$ are the intermediate ones. All the individuals have the same initial and final thicknesses.

4.2. Initial Population

To create the initial population an empirical expression for the maximum reduction is applied [26]. The maximum reduction to apply in each pass is given by (7), where n is the number of rolling passes in a mill train, h_0 is the entry thickness and h_f is the final thickness;

$$r = 1 - \sqrt[n]{\frac{h_f}{h_0}} \tag{7}$$

For each individual in the population the intermediate thicknesses ($h_1, h_2...h_5$) are determined, h_i it must be randomly generated in the interval $((1\text{-}r)h_i\text{-}1,\ h_i\text{-}1)$. Infeasible individuals don't satisfy the constraints given by (3) and (4). As we can see in this process feasible and infeasible individuals can be created. When a new individual is created its feasibility or unfeasibility is determined and saved.

4.3. Fitness Evaluation

The objective function of the problem is used, to calculate the fitness of each individual

4.4. Parents Selection

Two different criteria are applied to select the parents for crossover. When the population includes feasible and infeasible individuals, one feasible individual and one infeasible are uniform randomly selected. Otherwise if the population only includes feasible individuals we uniform randomly select one parent among the best individuals and one among the worst individuals.

4.5. Crossover Operator

Simulated Binary Crossover (SBX) is a crossover technique designed for real representations [24]. Once the parents P_1 and P_2 are randomly selected, the crossover operator produces the O_1 and O_2 offsprings. The SBX process consists in:

Step 1. Generate a uniformly distributed random number $u \in [0, 1]$
Step 2. Calculate the parameter β

$$\beta = \begin{cases} (2u)^{\frac{1}{n_c+1}} & \text{if } u \leq 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{\frac{1}{n_c+1}} & \text{otherwise} \end{cases}$$

Step 3. The offsprings are determined using the following vector crossover operation:

$$O_1 = 0.5[(P_1 + P_2) - \beta(P_2 - P_1)]$$
$$O_2 = 0.5[(P_1 + P_2) + \beta(P_2 - P_1)]$$

Whenever a parent is feasible and the other is infeasible, both offspringss are considered to substitute the parents, independently of the feasibility of the offsprings. Otherwise, when both parents are feasible, only feasible offsprings are considered. As we can see the genetic algorithm has two phases. In the first one the population contains feasible and infeasible individuals and in the second one includes only feasible individuals. The crossover operator is applied to a percentage of the population using an elitism criterion in the parent's substitution.

4.6. Mutation

In this process an individual of the population is uniform randomly selected and modified to produce a mutated individual, which is considered to substitute the original individual. Uniform mutation was applied [23]. An intermediate thickness h_i is uniform randomly selected, and a uniform random value in (h_i-1, h_i+1) is assigned to h_i. In the first phase, the mutation is applied to increase the population diversity. A mutated individual is considered to substitute the original individual, independently of its feasibility. In the second one only feasible mutated individual are considered.
A percentage of the population is mutated applying an elitism criterion in the original individual substitution.

4.7. Generations

A new generation is created applying crossover and mutation to the population of the current generation. The genetic algorithm stopping criterion is defined by a certain number of generations.

5. Experimental Results

The experiments were carried out with Microsoft Windows Server 2003 for Small Business, dual Xeon CPU 3.06 GHZ, 3.87 GB RAM, and the compiler C++. Industrial data were obtained from the Hylsa Monterrey Company and from the software HSMM of INTEG Process Group [13]. With this information 17 problem instances were defined, which can be consulted in [14]. Every instance is a different rolling problem where the intermediate reductions need to be found to roll the steel to obtain the final thickness in a 6-stand roll mill. The 17 instances considered were solved 30 times to obtain the average results. Preliminary tests were realized to obtain the best crossover and mutation values. The evaluated genetic algorithms were configured using a probability crossover of 40% and 50% of probability mutation, a population of 100 individuals and 100 generations. Four genetic algorithms evaluated were developed using different types of generating initial population strategy, crossover and mutation. Table 3 shows the algorithm configuration for each genetic algorithm evaluated.

Table 4 shows the accumulated average results for the 17 instances used. The first column contains the algorithm identifier. The column two shows the accumulated of the average execution time required for solving each instance. The third column contains the accumulated of the average execution time, required for the GA to get the best solution. The fourth column contains the average of the improvement percentage in the rolling time respect to the industrial time. As we can see the GA_4 algorithm shows the best performance.

Table 5 contains the rolling schedule proposed by the manufacturer, and the rolling schedule generated by the GA for the same instance. In both cases the exit thicknesses and the rolling time in each stand are showed.

Table 3. Evaluated genetic algorithms configurations

Algorithm Id	Initial Population	Crossover	Mutation
GA_1	Feasible individuals	Two points	Limited
GA_2	Feasible individuals	SBX	Limited
GA_3	Feasible individuals	SBX	Uniform
GA_4	Feasible and infeasible individuals	SBX	Uniform

Table 4. Efficiency and quality solution global results

	Total execution time (cpu. sec.)		Time to best solution (cpu. sec.)		Rolling time (seconds)
Algorithm	Average	Standard deviation	Average	Standard deviation	% Error
GA_1	306.169	22.913	233.680	21.4474	0.018%
GA_2	307.384	19.2202	245.761	28.1648	0.022%
GA_3	256.647	20.1310	240.673	20.816	0.045%
GA_4	42.625	4.9605	23.622	5.2910	0.051%

Figure 2 shows the typical differences observed between the rolling schedules proposed by the manufacturer and the genetic rolling schedules generated by the algorithm. The last one was graphed using the averages of the rolling time and of the thickness reductions in each one of the six stands for the 17 industrial instances used in this work.

Table 5. Rolling schedules for a 6-stand roll mill

Roll pass	Manufacturer proposed Exit thickness (mm)		Manufacturer proposed Rolling time (sec)	GA generated Exit thickness (mm)		GA generated Rolling time (sec)
	h_0	48		h_0	48	
1	h_1	26	4.433	h_1	36.67	4.401
2	h_2	14.3	2.494	h_2	22.52	2.498
3	h_3	9.31	1.603	h_3	13.57	1.610
4	h_4	6.03	1.043	h_4	8.23	1.046
5	h_5	4.55	0.769	h_5	5.80	0.771
6	h_f	3.8	0.0024	h_f	3.8	0.0039
	Total time :		10.3470	Total time:		10.3317

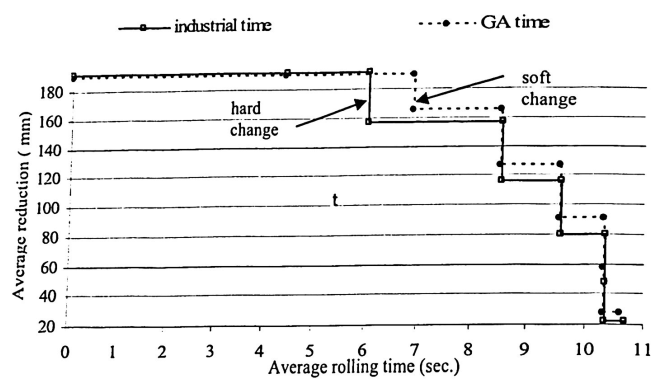

Fig. 2. Comparison of rolling schedule obtained with the GA_4 vs. the industrial schedule

The rolling time obtained with this algorithm was 0.051% better than the industrial time (Table 4). The execution average time required by the algorithm to solve an instance is 42 CPU seconds and 23 CPU seconds to obtain the best solution. Additionally the rolling schedule generated with GA_4 produces softer reductions than the rolling schedule proposed by the manufacturer (Figure 2). This characteristic of the solutions generated with GA_4 diminishes the equipment damage risks.

6. Conclusions

In this work the problem of minimizing the hot rolling time was approached. Unlike the traditional approaches, the steel chemical composition is incorporated as a problem parameter allowing the automatically setup of the hot mill for different steels. To validate the approach a six-stand rolling mill is modeled as an optimization constrained problem and an approach genetic algorithms based is proposed to solve the problem. Four different genetic algorithms were evaluated using realistic data of industrial schedules. The genetic algorithm configured with feasible and infeasible individuals in the initial population, SBX crossover and uniform mutation shows the best performance. The global rolling time obtained with this genetic algorithm was 0.051% better than the global industrial time and the average execution time required by the algorithm to obtain the best solution is 23.62 cpu seconds. Additionally the rolling schedule generated with the genetic algorithm diminishes the equipment damage risks because it produces softer reductions than the rolling schedule proposed by the manufacturer.

Currently we are developing new solution methods using different metaheuristics.

References

1. Dieter, G.E., Huhn, H. A., Semiatin, S. L., Chap 16-Rolling, *Handbook of Workability and Process Design*, ASM International, Materials Park, Ohio, 232 (2003).
2. Nolle, L., Armstrong, A., Hopgood, A., Simulated Annealing and Genetic Algorithms Applied to finishing mill optimization for hot rolling of wide steel strip *International Journal of Knowledge-Based Intelligent Engineering Systems*, 6(2), 104-111(2002).
3. Oduguwa, V., Tiwari, A., Sequential Process Optimisation Using Genetic Algorithms. *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg, 3242, , 782-791 (2004).
4. Chakraborti, N, Kumar, A., The Optimal Scheduling of a Reversing Strip mill: Studies using Multipopulation Genetic Algorithms and Differential evolution, *Materials and Manufacturing processes,*18(3), 433-445 (2003).
5. Chakraborti, N., Kumar, B., Babu, S., Moitra, S., Mukhopadhyay, A., Optimizing Surface Profiles during Hot Rolling: A Genetic Algorithm Based Multi-objective Optimization, Practical Approaches to Multi-Objective Optimization, *Comp. Mat. Sci.*, 37, 159-165(2006).
6. Chakraborti, N., Genetic algorithms in materials design and processing, *Int. Mat. Rev.*,49, (3-4), 246-260 (2004).
7. Gorni, A., The Application of Neural Networks in the Modeling of Plate Rolling Process, *JOM-e*, 49 (1997).
8. Nolle, L., Armstrong, A., Hopgood, A., Ware, A., Optimum Work Roll Profile Selection in the Hot Rolling of Wide Steel Strip using Computational Intelligence, *Lect. Notes in Comp. Sci.,* 1625, 435(1999).
9. Pataro, C., Herman, H., Direct determination of sequences of passes for the strip rolling by means of fuzzy logic rules. *Proc. 2d. Conf. on Intell. Proc. and Manuf. of Mat., IPMM.* 549(1999).

10. Shivpuri, R., Kini, S., Application of Fuzzy Reasoning Techniques for Roll Pass Design Optimisation. *Proc. 39th Mech. Work and Steel and Symp. on New Metal Form. Proc.* 755.(1998).
11. Macura, P., Petruska, J., Numerical and experimental simulation of pass rolling, *J. Mat. Pr. Tech.*, 60, 55(1996).
12. Wen, S., Hartley, P., Pillinger, I., Sturgess, C., Roll pass evaluation for three-dimensional section rolling using a simplified finite element method. *Proc. Inst. Mech. Eng.* 143(1977).
13. HSMM Rel. 3.0. *INTEG process group, inc.*, AISI/DOE Technology (2006).
14. Espriella, F. K.: *Optimización mediante algoritmos genéticos: aplicación a la laminación en caliente.* Master Thesis. Instituto Politécnico Nacional, CICATA Unidad Altamira (2008).
15. Medina, S. F., Hernández, C. A., General Expression of the Zener-Hollomon Parameter as a Function of the Chemical Composition of Low Alloy and the Microalloyed Steels. *Acta Metallurgica et Materialia*, 44(1), 137-148(1996).
16. Medina, S. F., Hernández, C. A., The Influence of Chemical Composition on Peak Strain of Deformed Austenite in Low Alloy and Microalloyed Steels. *Acta Metallurgica et Materialia,* 44(1),149-154 (1996).
17. Hernández, C. A., Medina, S. F., Ruiz, J., Modelling Austenite Flow Curves in Low Alloy and Microalloyed Steels. *Acta Metallurgica et Materialia*, 44 (1), 155-163 (1996).
18. Medina, S. F., Hernández, C. A., Modelling of the Dynamic Recrystallization of Austenite in Low Alloy and Microalloyed Steels. *Acta Metallurgica et Materialia,* 44 (1), 165-171 (1996).
19. Wusatowski, Z., *Fundamentals of Rolling*, Pergamon Press (1969).
20. Roberts, W., *Hot Rolling of Steel*, CRC; (1983).
21. Sims, R. B., The Calculation of Roll Forces and Torque in Hot Rolling Mill, *Proc. AIME* ,168(6), 191-200(1954).
22. Cook, P. M., McCrum, A. W., The Calculation of Load and Torque in Hot Flat Rolling, Reporte B.I.S.R.A., Londres (1958).
23. Ford, H., Alexander, J. M., Simplified Hot-Rolling Calculations, *J. Inst. Met.,* 92, 397-404 (1963).
24. Deb, K., Beyer, H.-G., Self-Adaptive Genetic Algorithms with Simulated Binary Crossover, *Technical Report No. CI-61/99*, Dept. of Computer Science/XI, University of Dortmund, March, 1-24 (1999).
25. Coello, C., Introducción a la computación evolutiva, CINVESTAV-IPN Departamento de Ingeniería Eléctrica, http://delta.cs.cinvestav.mx/~ccoello/compevol/ apuntes.pdf, mayo (2001).
26. Wright, H., Hope, T., Rolling of stainless steel in wide hot strip mills, *Metals Tech.*, 2 (12), 572-574(1975).