SIMULATED ANNEALING AND GENETIC ALGORITHMS APPLIED TO FINISHING MILL OPTIMISATION FOR HOT ROLLING OF WIDE STEEL STRIP

L. Nolle[1], D. A. Armstrong[2] , A. A. Hopgood[1] and J. A. Ware[3]

[1]Department of Computing and Mathematics, Nottingham Trent University
Burton Street, Nottingham, NG1 4BU, UK
Email: {lars.nolle|adrian.hopgood}@ntu.ac.uk

[2]Faculty of Technology, The Open University in Wales
24 Cathedral Road, Cardiff CF1 9SA, UK
Email: D.A.Armstrong@open.ac.uk

[3]School of Technology, University of Glamorgan
Treforrest, Pontypridd, CF37 1DL, UK
Email: jaware@glam.ac.uk

Abstract: *Genetic algorithms and simulated annealing have been evaluated for the optimisation of the set-up of the finishing train in a hot strip mill. Genetic algorithms proved unsuited to this optimisation problem, despite the development of a non-linear rank-based genetic algorithm that counters the effects of premature convergence and stalled evolution. The limitations of the genetic algorithm are believed to arise from epistasis, i.e. the interdependencies between input parameters. Simulated annealing, on the other hand, overcame this problem and was used for the final optimisation of the finishing train. The quality of the strip from a simulated mill was increased significantly as a result.*

Keywords: Rolling, Wide steel strip, Optimisation, Genetic algorithm, Simulated annealing, Rank-based, Epistasis.

1. Introduction

There is a world-wide overcapacity for wide steel strip. In such a "buyers' market", producers need to offer a high quality product at a competitive price in order to retain existing customers and win new ones. Producers are under pressure to improve their productivity by automating as many tasks as possible and by optimising the process parameters to maximise efficiency and quality. One of the most critical processes is the hot rolling of the steel strip [1].

2. Problem Domain

In a rolling mill a steel slab is reduced in thickness by rolling between two driven work rolls in a mill stand (Figure 1). To a first approximation, the mass flow and the width can be treated as constant. The velocity of the outgoing strip depends on the amount of reduction. A typical hot rolling mill finishing train might have as many as 7 or 8 close-coupled stands.

2.1 Hot Mill Train

A hot-rolling mill transforms steel slabs into flat strip by reducing the input thickness h_0 of approx. 200 mm to an output thickness h_7 of approx. 2 mm.

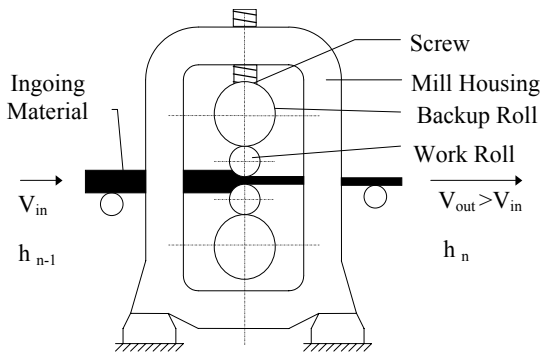


Figure 1- *Layout of a 4-high rolling mill stand.*

Figure 2 shows a typical hot strip mill train, consisting of a roughing mill (stands R1-R2) and finishing stands (F1 - F7). The roughing mill usually comprises one or more stands which may operate as a reversing mill, i.e. the slabs are reduced in

thickness in several passes by going through the mill in both directions. When the slab or plate has reached the desired thickness of approximately 35 mm it is rolled by the "close-coupled" finishing stands in one pass.

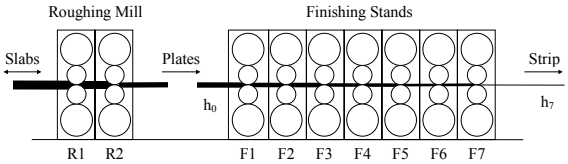
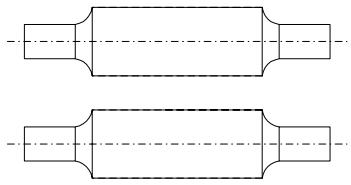


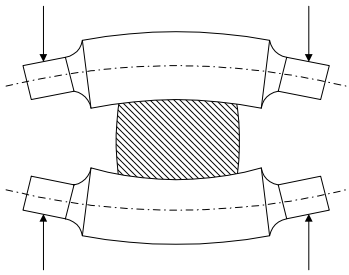
Figure 2- *Hot strip mill train.*

2.2 Factors Affecting the Strip Quality

The significant quality parameters of steel strip include: *dimensions*, *profile* and *flatness*. Strip profile is defined as variation in thickness across the width of the strip. It is usually quantified by a single value, the *crown*, defined as the difference in thickness between the centre line and a line at least 40 mm away from the edge of the strip (European Standard EN 10 051). Positive values represent convex strip profiles and negative values concave profiles. For satisfactory tracking during subsequent cold rolling a convex strip camber of about 0.5% – 2.5% of the final strip thickness is required [2]. Flatness – or the degree of planarity – is quantified in *I-Units*, smaller values of I-Units representing better flatness.



(*a*) - *unloaded rolls.*



(*b*) - *loaded rolls.*

Figure 3- *Factors affecting roll gap.*

Both the flatness and the profile of the finished strip depend crucially on the geometry of the loaded gap between the top and bottom work rolls. As a consequence of the high forces employed, the work rolls bend during the rolling process, despite being supported by larger diameter back-up rolls [3]. Figure 3a shows a pair of cylindrical work rolls. In Figure 3b the effects of the loading can be seen. If the geometry of the roll gap does not match that of the in-going strip, the extra material has to be accommodated by an increase in the width of the strip, known as spread, as well as an increase in its length (Figure 4). When the thickness is less than about 8mm, spread cannot take place and there is non-uniform elongation across the width, which can result in a wavy surface [4] (Figure 5).

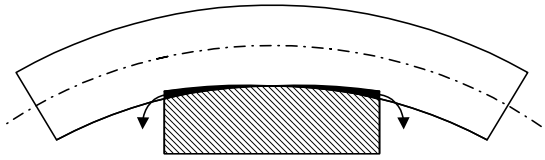


Figure 4 - *Mismatch of roll gap and strip geometry.*

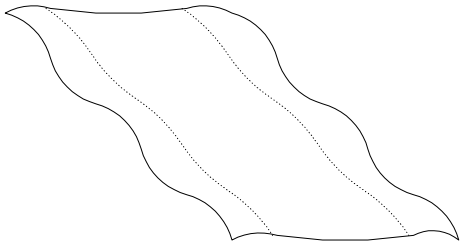


Figure 5 - *Wavy strip surface.*

3. Optimum Finisher Set-up

The challenge is to find the finisher set-up $h_1 ... h_7$, for a finishing mill train with seven mill stands, that will result in optimum strip quality, i.e. fitness. In this research, an existing computer model [5] has been used to simulate the finishing stands.

In this section, the experiments for the optimisation process, using *genetic algorithms* (GA) and *simulated annealing* (SA), are documented, and the results of the final optimisation are given.

3.1 The Optimisation Loop

Figure 6 shows the closed optimisation loop, containing the mill model and a suitable optimisation algorithm.

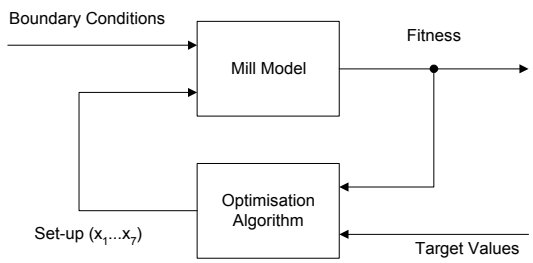
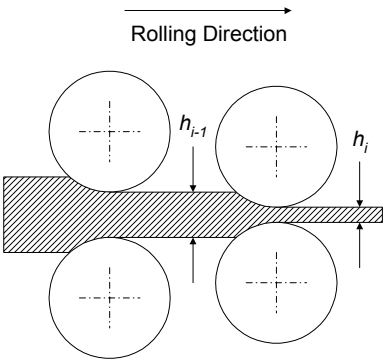
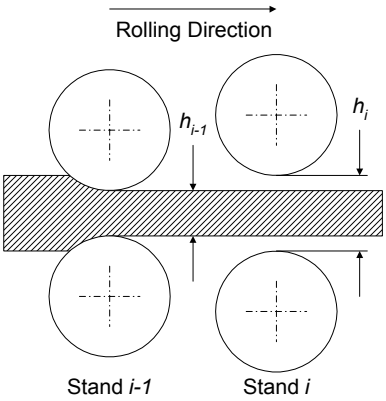


Figure 6 *- Optimisation loop.*

In a finishing train, the height h_i of the rolling gap of stand i must be less than the height h_{i-1} of the previous stand $i-1$ (Figure 7a). Otherwise, the rolls in stand i would not be in contact with the strip, and would leave stand i with a thickness of h_{i-1} (Figure 7b).



(*a*) - *Rolls in contact with strip.*



(*b*) - *Rolls not in contact with strip.*

Figure 7 *- Boundary condition.*

Hence, there is a boundary condition $h_i \geq h_{i+1}$. In order to satisfy this boundary condition with every possible candidate solution $\vec{x}$, the parameter *reduction* $r \in [0,1]$ has been used in preference to *thickness* h as an input parameter for the mill. The reduction r_i of stand i can be calculated as follows:

$$r_i = \frac{h_i}{h_{i-1}} \tag{1}$$

where:

r_i	reduction of stand i,
h_{i-1}	thickness of ingoing material (i.e. height of the roll gap in previous stand),
h_i	gap height of stand i.

The overall reduction $r_{overall}$ is given by the desired strip thickness h_7 after the last mill stand and the thickness h_0 of the in-going plate:

$$r_{overall} = \frac{h_7}{h_0} \tag{2}$$

Each element of the output vector $\vec{x} = (x_1, x_2, \ldots, x_7)$ of the optimisation algorithm must be re-scaled by multiplication with a factor c in order to match the desired overall reduction:

$$r_{overall} \equiv \prod_{i=1}^{7} r_i = \prod_{i=1}^{7} c x_i \tag{3}$$

This factor c must be re-computed for every new candidate solution of the optimisation algorithm by using equation 4:

$$c = \left[\frac{r_{overall}}{\prod_{i=1}^{7} x_i} \right]^{\frac{1}{7}} \tag{4}$$

It can be seen that each input variable affects the relative weight of all the other input variables, because the actual thickness h_i of stand i depends not only on the reduction r_i, but also on the reduction in each of the previous stands. This parameter interaction, known as *epistasis* [6], provides a challenging optimisation problem.

3.2 The Fitness Function

The fitness (objective function) used is a combination of crown and flatness values of the centre-line, the edge, and the quarter-line (equation 5). To avoid a division by zero, 1 has been added to the denominator.

$$f(x,\alpha) = \frac{1}{1 + \frac{1}{\alpha}\sum_{i=1}^{3} I_i(x) + \left|c_{aim} - c(x)\right|} \tag{5}$$

where:

$f(x, \alpha)$	fitness of solution x, given parameter α,
α:	constant to determine the relative contribution of flatness and camber, chosen to be 5000 for the experiments,
$I_i(x)$	I-Units at line i for solution x,
c_{aim}	target crown,
$c(x)$	achieved crown for solution x.

Because the flatness values are about 5000 times greater than the differences between the target crown and the achieved camber, the constant α has been chosen to be 5000 for the experiments. This is to ensure the same order of magnitude for the achieved flatness and the achieved strip camber, i.e. to accord both parameters the same importance.

The theoretical maximum value of this objective function is 1.0 but, due to the non-deterministic influence of the simulated measurement devices, the target maximum of this function is 0.994. The lowest acceptable value of 0.969 is determined by the tolerances defined by the client.

3.3 Evaluation of Optimisation Algorithms

Two experimental optimisation algorithms have been examined: *simulated annealing* (SA) and *genetic algorithms* (GA).

3.3.1 Genetic Algorithms

As Darwin's theory of natural selection articulates, nature is very effective at optimisation by means of the simple method of trial and error, e.g. enabling life-forms to survive in an unfriendly and changing environment. Genetic algorithms simulate this evolutionary mechanism by using heredity and mutation. They were first introduced in 1975 by Holland [7] who also provided a theoretical framework for Genetic Algorithms, the Schemata Theorem [8].

For a genetic algorithm, all input parameters to a system are encoded as a binary string, the so-called *genotype*. The individual represented by the genotype is called a *phenotype*. This phenotype needs a certain quality or fitness to survive. The search is now undertaken by a population of N genotypes, i.e. the genepool, rather than a single individual. Therefore, the search space is tested at N points in parallel.

The population of individuals of the genepool at a time t_i is called a generation. A new generation for time t_{i+1} can be generated by selecting fit individuals (with a fitness above the average fitness of the genepool) from the current population and recombining them with a probability p_c by using the crossover operator and applying mutation with the probability p_m. The offspring are inserted into the new generation genepool. The search is usually carried out until either a fitness threshold or a limit on the number of iterations has been reached.

In these experiments, a variation of GA has been used, a rank-based genetic algorithm (RBGA) [9]. Figure 8 shows the flow chart of RBGA. It uses a non-linear selection function (equation 6), which was developed in earlier work [10]. It has been demonstrated that such a GA outperforms other GAs in this problem domain [5]. The non-linear selection function $n(x,c)$ scales the exponential function so that for given input values $x \in [0,1]$ the output is between 0 and N:

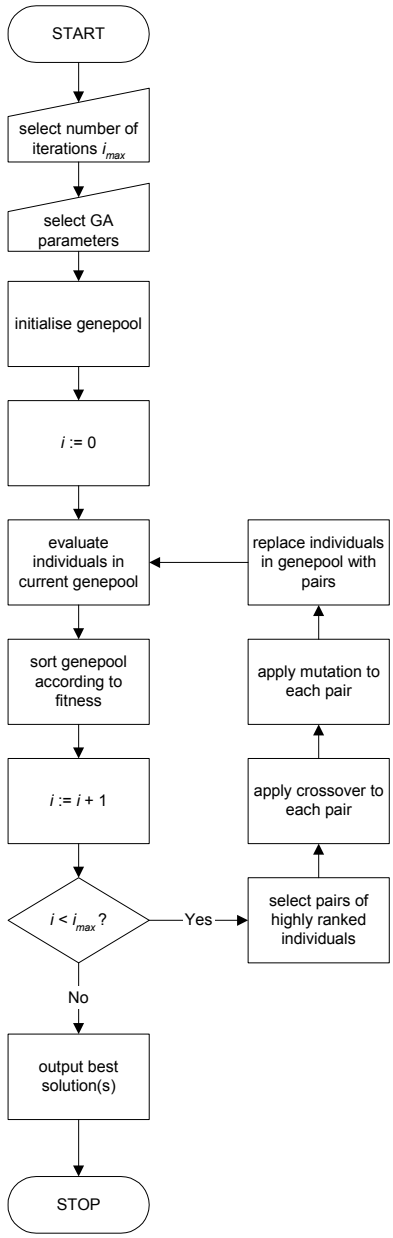


Figure 8 *– Flowchart of the basic RBGA.*

$$n(x,c) = ceil\left[\frac{N}{1-\frac{1}{e^{c}}}\left(1-\frac{1}{e^{cx}}\right)\right] \qquad (6)$$

where:

n	individual chosen from the current genepool to join the mating pool,
N	number of individuals in genepool,
x	equally distributed random number [0, 1],
c	constant which controls the non-linearity of the function,
$ceil(a)$	function returning the smallest integer that is not less than its argument a.

In contrast to the original Rank Based GA published by Baker [9], the best individual becomes the highest rank in this case. The degree of non-linearity can be controlled by changing the constant c. This allows on-line adaptation of the algorithm. For example, c could be adjusted to the ratio (on-line performance/off-line performance).

For the experiments, uniform crossover [11] and two different crossover probabilities p_c have been used with the RBGA.

3.3.2 Simulated Annealing

This general optimization method was first introduced by Kirkpatrick et. al. [12], based on the work of Metropolis et. al. [13]. It simulates the softening process ("annealing") of metal. The metal is heated-up to a temperature near its melting point and then slowly cooled-down. This allows the particles to move towards an optimum energy state, with a more uniform crystalline structure. The process therefore permits some control over the microstructure.

Simulated annealing (SA) is a variation of the hill-climbing algorithm. Both start off from a randomly selected point within the search space. The difference between them is that if the fitness of a new trial solution is less than the fitness of the current solution, the trial solution is not automatically rejected, as in hill climbing. Instead it becomes the current solution with a certain transition probability $p(T)$, which depends on the difference in fitness and the temperature (Equation 7). Here, "temperature" is an abstract control parameter for the algorithm rather than a real physical measure. The transition probability $p(T)$ for a given temperature and a given difference in fitness ΔF can be determined as follows:

$$p(T) = e^{-\frac{\Delta F}{T}} \tag{7}$$

where:

$p(T)$	transition probability in the interval [0,1],
ΔF	reduction in fitness of the trial solution compared with the current solution ,
T	current temperature.

In Figure 9, the graphical representation of Equation 6 can be seen for expected values of ΔF . In order to achieve an initial transition probability of 0.5, the start temperature T_0 has been chosen to be 0.2.

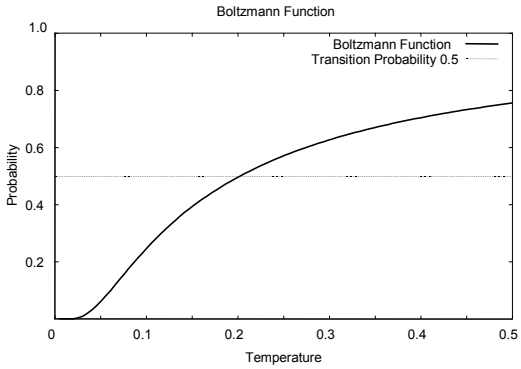


Figure 9 - *Boltzmann function.*

The algorithm starts with a high temperature, which is then reduced in steps:

$$T_{t+1} = \alpha T_t \tag{8}$$

where:

T_{t+1}	next temperature,
T_i	current temperature,
α	cooling coefficient.

On each step, the temperature must be held constant for an appropriate period of time (i.e. for an appropriate number of iterations) in order to allow the algorithm to settle into a "thermal equilibrium", i.e. a balanced state. If this time is too short, the algorithm is likely to converge to a local minimum. The combination of temperature steps and cooling times is known as the "annealing schedule", which is usually selected empirically.

Figure 10 shows the flowchart of the basic SA algorithm. The coefficient α has been chosen to be 0.99 in order to approach a probability of about zero after approximately 4000 iterations. The number of iterations for one temperature step has been chosen to be 50. This number was determined by experiments: smaller numbers resulted in worse solutions, larger numbers only increased the computation time.

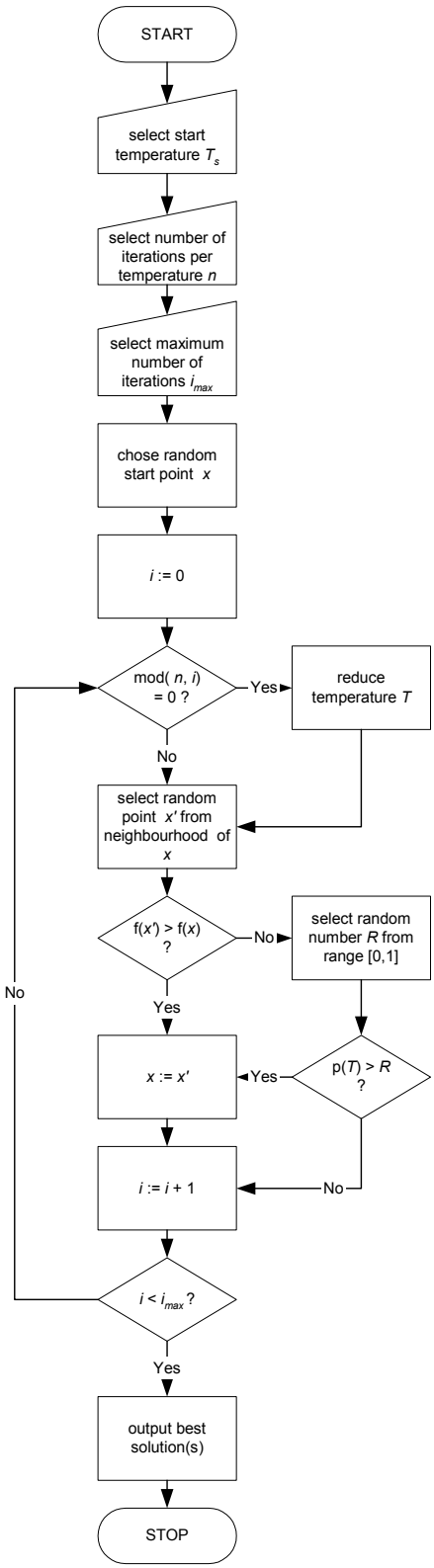


Figure 10 *– Flowchart of the basic SA algorithm.*

3.3.3 Comparison between algorithms

In order to compare the methods, each algorithm has been applied 50 times. To allow for a fair comparison of the different algorithms, the term *guess* has been used rather than *generation*: one guess equals one model calculation, e.g. a GA with a genepool containing 50 individuals carries out 500 guesses in 10 generations. Each run has been allowed 4000 guesses.

Table 1 gives the results for the average fitness found by each method, and the average number of guesses taken by the method to find the best result during one run.

Method	Average Fitness	Standard Deviation of Fitness	Average Number of Guesses	Standard Deviation of Guesses
RBGA (p_c=0.6)	0.962514	0.000825	3636.8	387.1
RBGA (p_c=0.0)	0.962682	0.001078	3332.0	597.5
SA	0.980136	0.012214	3108.6	897.4

Table 1 - *Experimental Results.*

It can be observed that the RBGA has achieved an average fitness of about 0.962 in both cases, despite the very different crossover probabilities. Hence, the crossover operator has no effect on the search results.

The solutions found by the different methods have then been divided into two classes. The first class ("unsuitable solutions") contains all solutions performing below the lowest acceptable tolerance limits (fitness < 0.969, see Section 3.2). The second class ("good solutions") contains all suitable solutions, i.e. those with a fitness over the lower tolerance limits.

Method	Unsuitable Solutions	Good Solutions
non-linear Rank Based GA (p_c=0.6)	100%	0%
non-linear Rank Based GA (p_c=0.0)	100%	0%
Simulated Annealing	4%	96%

Table 2 - *Class analysis.*

It can be seen from Table 2 that 96% of the solutions found by SA are within the defined tolerances, i.e. are usable solutions. In contrast, none of the solutions found by GA reaches the required fitness. Hence, GA failed completely in this application. This is caused by the nature of the optimisation problem, i.e. the high degree of

epistasis present in the system. Figure 11 and Figure 12 show that only the convergence speed is affected by crossover. Due to the dependent coding of the input parameters, the GA has been degenerated into random search.

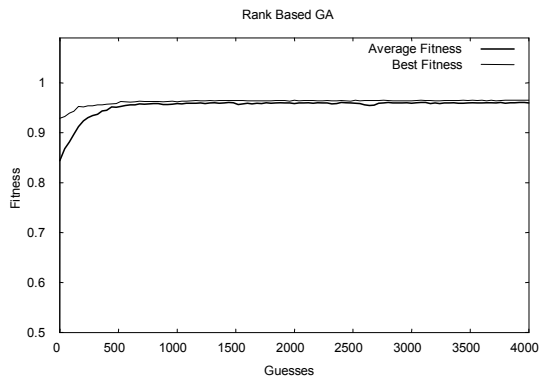


Figure 11 - *Typical run of the RBGA with a cross-over probability of 0.6.*

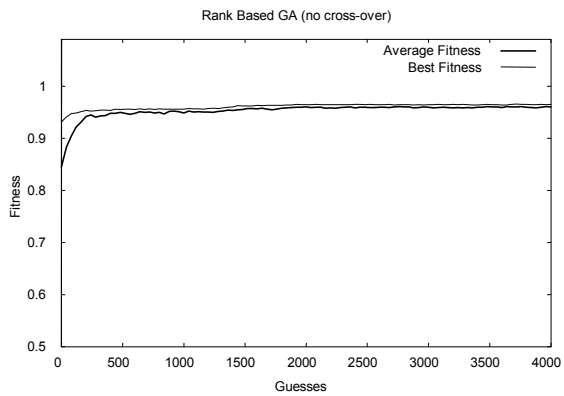


Figure 12 - *Typical run of the RBGA with a crossover probability 0.0.*

Figure 13 shows a typical run of SA. Figure 15 shows that SA has found a solution in almost every run. The points represent the average achieved fitness, the length of the error bars the standard deviation.

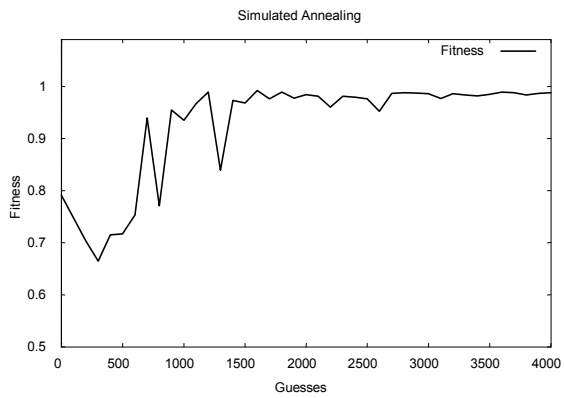


Figure 13 - *Typical run of the SA algorithm.*

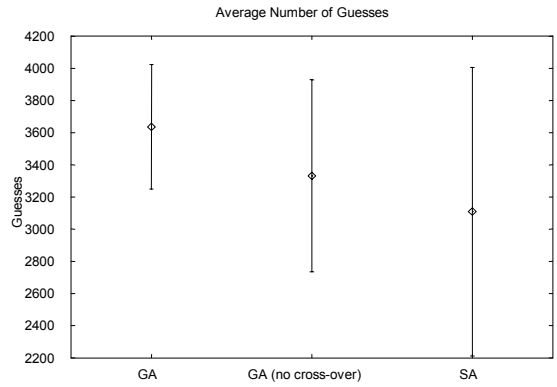


Figure 14 *- Average number of guesses.*

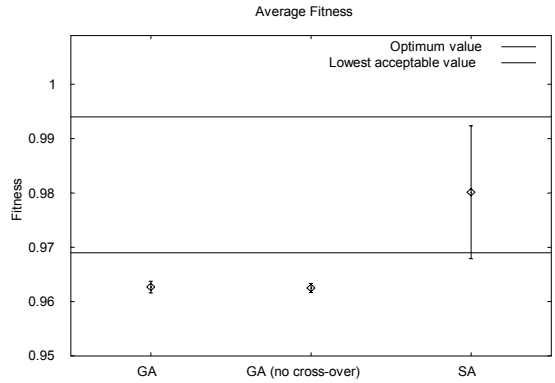


Figure 15 *- Comparison of fitness results.*

Based on these results, SA has been chosen for the final optimisation.

4. Final Results

In these experiments, the finishing mill set-up has been optimised for five different slabs. The start temperature T_0 had been chosen to be 0.2. The annealing function was $T_{i+1} = 0.99T_i$ and the number of iterations per temperature was 50. The maximum number of guesses was 5000. Table 3 gives the details of the results.

It can be seen that in all cases considerable improvements in flatness have been achieved, although the solutions found by the algorithm did not differ greatly from the original settings. An average improvement of 66.14% has been accomplished.

Slab No.	Original Flatness [I-Units]	Improved Flatness [I-Units]	Improvement [%]
1	194.67	9.00	95.4
2	21.33	10.33	51.6
3	6.00	3.67	38.8
4	26.00	5.33	79.5
5	27.00	9.33	65.4

Table 3 - *Comparison of original and improved flatness.*

5. Conclusion

For this particular optimisation problem, two experimental optimisation algorithms have been compared: simulated annealing and a non-linear rank-based genetic algorithm.

The failure of genetic algorithms in this optimisation problem is due to epistasis, the interacting of the input parameters, i.e. one input parameter has a strong influence on the other input parameters. Simulated annealing performed demonstrably better and was therefore used to optimise the finisher set-up for five different steel slabs. The quality of the strip profile and flatness from the simulated mill was significantly increased. It has been demonstrated that simulated annealing is capable of solving optimisation problems of a physical system, even where a dependency exists between the inputs.

Acknowledgements

This work was funded by *Metrology Systems Wales, Cowbridge, UK*.

References

[1] Larke, E. C.: The Rolling of Strip Sheet and Plate, Chapman and Hall Ltd, London, 1963

[2] Winkler, W.: Grundlagen des Breitbandwalzens, Stahl u. Eisen 63 (1943) Nr. 40, pp 731-735

[3] Emicke, Lucas: Einflüße auf die Walzgenauigkeit beim Warmwalzen von Blechen und Bändern, Neue Hütte, 1. Jg. Heft 5, 1956, pp 257-274

[4] Wilms, Vogtmann, Klöckner, Beisemann, Rohde: Steuerung von Profil und Planheit in Warmbreitbandstraßen, Stahl u. Eisen 105 (1985) Nr. 22, pp 1181-1190

[5] Nolle, L., Armstrong, D.A., Hopgood, A.A., Ware, J.A.: Optimum Work Roll Profile Selection in the Hot Rolling of Wide Steel Strip Using Computational Intelligence, Lecture Notes in Computer Science, Vol. 1625, Springer, 1999, pp 435-452

[6] Davidor, Y.: Epistasis Variance: Suitability of a Representation to Genetic Algorithms, Complex Systems 4 (1990), pp 369-383

[7] Holland, J.H.: Adaptation in Natural and Artificial Systems, University of Mitchigan Press, 1975

[8] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Lerning, Addison-Wesley, 1989

[9] Baker: Adaptive Selection Methods for Genetic Algorithms, Proceedings of an International Conference on Genetic Algorithms and Their Application, Hilldsdale, 1985, pp 101-111

[10] Nolle, L., Armstrong, D. A., Lee., S.: On a Class of Non-linear Rank Based Genetic Algorithms, Proceedings of the 6th International MENDEL Conference on Soft Computing, Brno, CZ, 7-9 June 2000, pp 101-106

[11] Syswerda: Uniform Crossover in Genetic Algorithms, Proceedings of the International Conference on Genetic Algorithm 1989 (ICGA'89), pp 2-9

[12] Kirkpatrick, S., Gelatt, C.D., Jr., Vecchi, M.P.: Optimization by Simulated Annealing, Science, 13 May 1983, Vol. 220, No. 4598, pp 671-680

[13] Metropolis, A., Rosenbluth, W., Rosenbluth, M.N., Teller, H., Teller, E.: Equation of State Calculations by Fast Computing Machines, The Journal of Chemical Physics, Vol. 21, No. 6, June 1953, pp 1087-1092