ELSEVIER

Applied Mathematical Modelling 25 (2001) 287–297

APPLIED MATHEMATICAL MODELLING

www.elsevier.nl/locate/apm

A study of the continuous casting mold using a pareto-converging genetic algorithm

N. Chakraborti [a,*], R. Kumar [b], D. Jain [c]

[a] *Department of Metallurgical and Materials Engineering, Indian Institute of Technology, Kharagpur 721 302, India*
[b] *Department of Computer Science & Engineering, Indian Institute of Technology, Kharagpur 721 302, India*
[c] *Chemical Engineering Group, Birla Institute of Technology and Science, Pilani, Rajasthan 333 031, India*

Received 6 November 1998; received in revised form 25 January 2000; accepted 29 August 2000

Abstract

The mold region of the continuous caster, the most widely used casting device used by the steel industry has been modeled through a combination of a steady-state heat transfer approach and a recently developed pareto-converging genetic algorithm (PCGA). Due to highly non-linear nature of the objective functions, as well as the constraints, locating the pareto-front was quite a challenging job in this case. Also, from a physical consideration, the pareto-front needed to be zoomed into the region of equality of two objective functions. PCGA could successfully locate the optima after an extensive search, and the predictions are well in accord with the data provided by a number of industrial casters. © 2001 Published by Elsevier Science Inc. All rights reserved.

1. Introduction

Continuous casting technology is a major breakthrough in the history of steelmaking and is immensely influencing the steelmaking practice worldwide. In fact, it has already rendered traditional ingot casting virtually redundant in a number of countries where the major steel companies are basically moving towards hundred percent production through the continuous casting route. The basic features of a continuous caster are shown in Fig. 1. The molten steel from a *tundish* is poured into the caster where the primary solidification occurs in a water-cooled mold. The metal subsequently passes through a spray-cooling region where the secondary cooling occurs.

Application of biologically inspired genetic algorithms (GA) [1,2] to continuous casting has just begun. Earlier it was demonstrated by Chakraborti et al. [3,4] how the casting velocity can be maximized in terms of various parameters in the mold region. GA techniques related to continuous casting have recently been implemented in a steel plant in Slovenia by Filipic and Sarler [5] which clearly indicates the emerging importance of GA applications in this area. The present paper reports an extension of our earlier work [4] using a *pareto-optimal* formulation [6] and a

* Corresponding author. Tel.: +91-3222-83286; fax: +91-3222-82280.
E-mail address: nchakrab@metal.iikgp.ernet.in (N. Chakraborti).

0307-904X/01/$ - see front matter
PII: S 0 3 0 7 - 9 0 4 X (0 0) 0 0 0 4 7 - 0

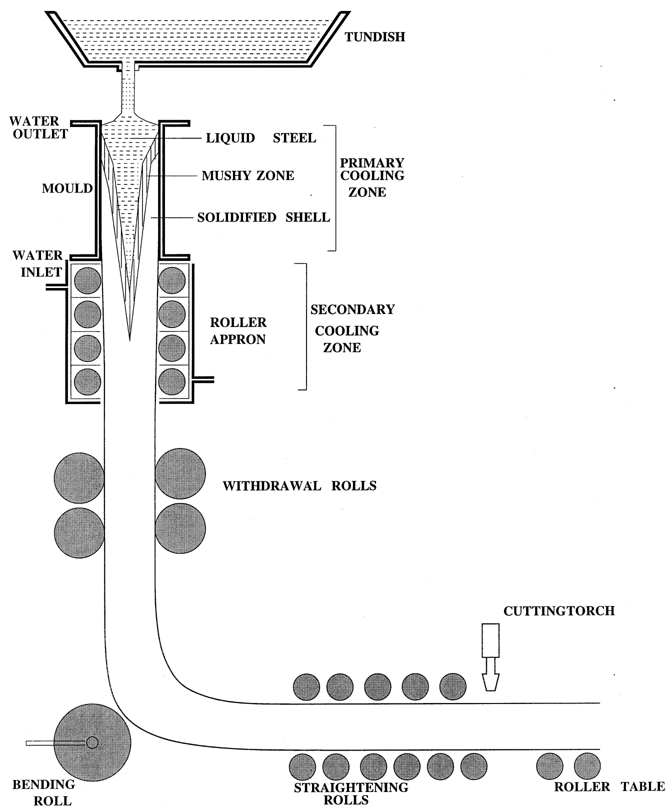

Fig. 1. A schematic digram of the continuous casting process.

pareto-converging genetic algorithm (PCGA) recently developed by Kumar [7]. Further details are provided below.

2. Scope of the present work

The present study focuses on the mold region, the very heart of the caster, where proper control of the solidification process goes a long way towards the final quality of the cast product. Despite years of both theoretical and experimental research [8,9] the process parameters pertinent to this region of the caster have really not been *optimized*, at least in the mathematical sense of the word. This is reflected in the wide variation in the operational parameters used by the industrial casters surveyed by Samarasekera et al. [10], many of which are attempting to cast billets of similar cross sections under significantly different conditions. In fact, highly empirical practices abound in the industrial scene. Placing of a coin by the mold side to detect the mold oscillation, as described by Samarasekera et al. [10], is perhaps an extreme example in this context. In addition, there are plenty of other methods of parameter adjustments which may not be so much outrageously unscientific, but nonetheless equally ad hoc in nature. In this scenario, there remains an ample scope of conducting a mathematical optimization of the continuous casting mold operation, which till to date have been performed only to a limited extent. A probable reason for this is perhaps a highly non-linear and often multi-modal nature of the problem, rendering numerical computation immensely difficult, particularly by the derivative based traditional methods.

In our earlier work [4], an attempt was made to conduct an optimization study of the mold region using a number of computational techniques, among which genetic algorithms had turned out to be the most promising. Since any industrial caster would like to cast as fast as possible maintaining an acceptable limit for the maximum shell thickness, the basic philosophy behind our optimization scheme was to maximize the casting velocity subject to various system constraints. In order to achieve that a number of system variables were identified, and attempts were made to express the casting velocity in terms of these variables, utilizing a number of physical criteria – a steady state energy balance for example. It was however, not possible to incorporate all the variables in a single equation. More than one objective functions were therefore necessary to describe the system, and in our previous work [4] those were handled according to the *Method of Objective Weighing*, where in order to maximize or minimize $f_i(\chi)$ for $i = 1, 2, \ldots, N$ (χ being an M dimensional, vector containing M design variables) one optimizes the overall objective function $\Phi = \sum_{i=1}^{N} \tilde{w}_i f_i(\chi)$, such that $0 \leqslant \tilde{w}_i \leqslant 1$ and $\sum_{i=1}^{N} \tilde{w}_i = 1$.

Although convenient to use and also known as a classical approach towards handling the multi-objective problems, the Method of Objective Weighing is often unable to offer justice to them. In many real world problems the weightage factor ϖ_i is difficult to assign and various objective functions are not necessarily correlated in a linear fashion, as implicitly assumed in this method. A much better approach would be resorting to a pareto-optimal formulation where each objective function is computed separately and various sets of solutions are compared with each other to identify a series of feasible solutions, constituting a so-called *pareto-front*, where no member is absolutely superior over any other. This is ensured by implementing a condition of *non*-dominance; stripped of the mathematical jargon which essentially means that no solution is accepted in the pareto set which is either comparable or better than its other members, in terms of *all* the objective functions, and is better in terms of *at least one*. A pareto-optimal formulation provides an engineer with an ample choice of the decision variables and has been tested in hosts of problems from various disciplines [11–13]. However, to the best of our knowledge, no pareto-optimal studies have been reported for the continuous casting process. This paper perhaps is the maiden attempt to introduce this powerful design tool into continuous casting research. The version of genetic algorithm (i.e. PCGA) that has been used to achieve this is also quite new and has recently been proposed by one of the present authors [7]. Implementation of the pareto formulation in the present context is far from an easy job. In addition of the inherent non-linearity of the problem, handling more than one complicated search space simultaneously appeared to be quite a challenging task for the genetic algorithm employed by us. Applying a too complicated system model at this stage would have simply resulted in a computational deadlock, serving no rational purpose at all. After a judicious deliberation what we have employed here is a relatively simple but nonetheless efficient model of the casting process. Although we significantly altered and fine tuned it to fit our requirements, the fundamentals of this model have been available in the metallurgical literature for quite some time [14]. Surprisingly, however, it remained grossly unutilized as far as design oriented research is concerned in the continuous casting area. This study therefore is an attempt to introduce a newer approach both in terms of modeling and computing methodology. Further details are provided below.

3. The pareto-optimal formulation

As indicated earlier, we will restrict our attention to the mold region only, where the primary cooling of the steel takes place in a water cooled copper mold supported by steel baking plates [9]. Following Brimacombe and Samarasekera [8] we will assume the mold configuration shown in

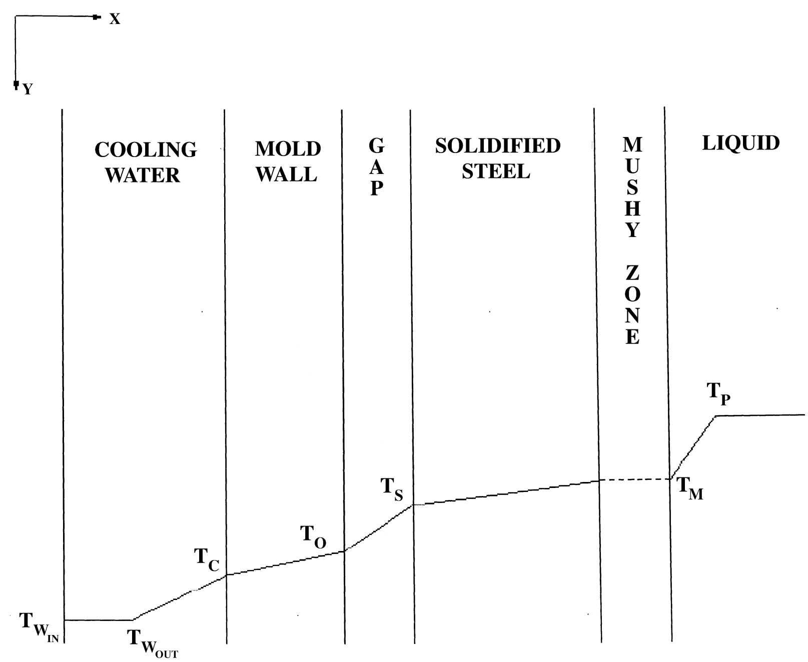

Fig. 2. A schematic diagram of the configuration of the mold region.

Fig. 2. Following now a procedure suggested by Geiger and Poirier [14] one can perform a series of steady state heat balance for the present configuration to obtain an expression for the first objective function, which is essentially casting velocity u expressed as:

$$u = \frac{2k'(\hbar + \delta_1)(T_M - T_o)L}{\rho' H'_f a[2k'M + M^2(\hbar + \delta_1)]}, \tag{1}$$

where k' is the thermal conductivity of the metal, $\hbar$ is an average heat transfer coefficient for the mold–metal interface, δ_1 is a correction factor, T_M and T_o denote temperatures at the solid–liquid interface and the inner surface of the mold wall respectively, L is the mold length, M is the solidified mold thickness, ρ' is the density of the steel and the lumped parameter a is defined as:

$$a = \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{C'_P(T_M - T_o)}{3[(1-\varepsilon)H'_f]}}. \tag{2}$$

The latent heat of fusion was adjusted through a correction factor ε in order to account for the heat retention in the *mushy zone* that forms between the liquid and solid (Fig. 2) and was defined as:

$$H'_f = (1-\varepsilon)[H_f + C_{PL}(T_P - T_M)], \tag{3}$$

where H_f is the uncorrected latent of fusion, C_{PL} is the specific heat of liquid and T_P is the pouring temperature of the liquid metal.

In this study, Eq. (1) was taken as a first objective function which was maximized subject to the following constraints arising out of a number of physical considerations. By equating the average mold heat-flux expression provided by Lait et al. [15] with the heat-flux at the mold–metal interface, the first physical constraint was formulated as

$$2675.2 - 334.4\sqrt{\frac{L}{u}} - (\hbar + \delta_1)(T_S - T_o) = 0, \tag{4}$$

where the time was scaled as L/u and the original constants were converted to SI units. The mold-gap of thickness Δg is generally filled with mold powder of thermal conductivity k_m a second equality constraint was therefore worked out through a simple flux continuity in the gap region, such that:

$$\frac{k_m + \delta_2}{\Delta g} - (\hbar + \delta_1) = 0. \tag{5}$$

Due to a significant uncertainty in the reported values of k_m, a correction factor δ_2 was added to it. Finally, the extent of heat removal from the mold is limited by the extent of heat removal through forced convection in the cooling water channel. Conducting a heat balance at the cooling water side a third constraint was therefore formulated as

$$h_w k[(T_o - T_{Wout})\{\rho_w C_{PW}\phi(T_{Wout} - T_{Win}) - \beta\}(k + \Delta t_{mold} h_w)] = 0, \tag{6}$$

where h_w is the heat transfer coefficient at the cooling water side, k the thermal conductivity of copper mold, ρ_w and C_{PW} are the density and specific heat of the cooling water, while T_{Wout} and T_{Win} are its inlet and outlet temperatures, respectively. The cooling water velocity is denoted by ϕ, β is a water heat loss adjustment term and Δt_{mold} is the thickness of the mold.

In the final formulation, the equality constraints were combined to yield

$$\Im - \Gamma[h_w k(T_S - T_{wout}) - \{\rho_w C_{PW}\phi(T_{Wout} - T_{Win}) - \beta\}(k + \Delta t_{mold} h_w)] = 0, \tag{7}$$

where

$$\Im = 2675.2 - 334.4\sqrt{\frac{L}{u}} \tag{8}$$

and

$$\Gamma = \frac{(k_m + \delta_2)}{\Delta g h_w k}. \tag{9}$$

Further details of these equations are provided elsewhere [16].

The continuous casting mold however oscillates with a sinusoidal motion, which has not been accounted for in the equations given above. In terms of mold oscillation the casting velocity can be expressed as [17]:

$$u = \pi f S \cos(\pi f t_N), \tag{10}$$

where t_N denotes *negative-strip time*, the time period in the mold oscillation cycle in which the downward velocity of the mold exceeds that of the strand, and f and S are the frequency of oscillation and its corresponding stroke length, respectively.

Since the two expressions for the casting velocity are composed of variables independent of each other, one needs to maximize them simultaneously which leads to a pareto-optimal problem. Fixing of the casting velocity based upon just one of them would lead a false operating guideline. Furthermore, the velocities predicted by both the objective functions should match reasonably which is often difficult to attain computationally. Considering the fact that the objective functions belong to two different search spaces, and most of the parameters involved in them, in reality, can vary within prescribed limits, the optimum velocity is not possible to calculate without resorting to a robust optimizing technique. In fact, during our earlier work [4] it was found that the gradient-based Steepest Descent method failed miserably to optimize such objective functions when all the variables were taken into account, while genetic algorithms were able to produce acceptable results. It also needs to be pointed out that both the objective functions (i.e. the expressions of casting velocity) here are highly non-linear in nature. In addition, a non-linear reverse dependence of the system constraint on the casting velocity makes this problem additionally complicated more than many other similar non-linear systems. [1] Here the final equality constraint (Eq. (3)) was handled using a *parabolic-penalty parameter* approach [18], which in the present scenario was not very easy to implement. Furthermore, since both the objective functions provide the value of the same variable u, here the pareto-front needs to be zoomed into the range where the predicted values from both the equations are practically equal. The two expressions however belong to two unrelated and often conflicting search spaces, with entirely different *fitness landscapes*, as it is known in the GA literature [1,2]. This zooming in operation was therefore a highly cumbersome task, which was ultimately quite successfully handled by our algorithm. The details of our PCGA technique are provided next.

4. Pareto-converging genetic algorithm

The PCGA has been recently proposed by Kumar [7] for the optimization of multi-objective functions. This can be considered to be an extension of the earlier work of Fonseca and Fleming [19] and also of Srinivas and Deb [20]. The basic methodology of PCGA is described below.

PCGA uses the basic operators of Simple Genetic Algorithm, as detailed in our earlier work [16]. However, instead of a single set of *population* traditionally used in SGA, here more than one set of randomly initiated populations is used. Each of them is known as a *tribe* in the PCGA terminology. The genetic operations like *crossover*, *mutation* etc. are carried out within the individual tribes, and intermixing of tribes is allowed only for convergence checking. Like the technique suggested by Srinivas and Deb [20] PCGA also resorts to *non-dominated sorting* of the population set. For a typical multi-objective problem:

$$\text{Minimize Objective } f_m(\vec{\chi}), \quad m = 1, 2, \ldots, M$$
$$\text{Subject to Constraint } g_k(\vec{\chi}) \leqslant ck, \quad k = 1, 2, \ldots, M,$$

where $\vec{\chi} = (x_n\colon n = 1, 2, \ldots, N)$ is a N-tuple vector of variables; and $\vec{\Phi} = (f_m\colon m = 1, 2, \ldots, M)$ is a N-tuple vector of objectives.

If the objective vector $\vec{\Phi}_i$ is *partially* less than another objective vector $\vec{\Phi}_j$, $(\vec{\Phi}_i \prec \vec{\Phi}_j)$ then $\vec{\Phi}_i$ is taken to be dominating over $\vec{\Phi}_j$. Following Goldberg [1] the condition for being partially less is taken as:

$$\left(\vec{\Phi}_i \prec \vec{\Phi}_j\right) \Longleftrightarrow (\forall_m)\left(f_{mi} \leqslant f_{mj}\right) \wedge (\exists_m)\left(f_{mi} < f_{mj}\right). \tag{11}$$

The locus of the non-dominated solutions constitutes the pareto-front.

In PCGA all the *individuals* are *ranked* based upon their *non-dominance.* The procedure of ranking is already elaborated in the GA literature [12]. A simple linear function is utilized to map all the N individuals of a certain tribe onto some dummy fitness values, based upon which a simple *roulette wheel* selection [18] was conducted for picking up the pair of mates for *crossover* and *mutation.* After crossing over and mutating those mates, the newly produced *offsprings* are

[1] Since u is also a parameter present in the constraint expression shown in Eq. (7), here the constraint is non-linearly dependent on the velocity as well, not just the other way round. Optimizing $u = f(u)$ for a non-linear function, as needed in this case, is quite a challenging job from a mathematical point of view.

sent back to the tribe, temporarily increasing its size to $N+2$. The two children produced this way could be of better rank than their parents; alternately there could be the so-called *lethal* reproductions as well, giving rise to offsprings of inferior ranks. To resort back to the original size, the population is then ranked again, and the two individuals with lowest ranking are discarded. When $N/2$ pair such pair-wise reproductions are completed, the PCGA is said to finish an *epoch*, in contrast to a *generation* of SGA, where the offsprings are not pitted against the parent population. [2]

Unlike Srinivas and Deb [20] no *sharing* was attempted here. Instead, the above-mentioned process was repeated till convergence. After any epoch t, a rank ratio $\Re$ for any particular rank was calculated as:

$$\Re = \frac{\mathrm{pop}(t)}{\mathrm{pop}(t) + \mathrm{pop}(t-1)}, \tag{12}$$

where $\mathrm{pop}(t)$ denotes the number of individuals belonging to that rank at epoch t and $\mathrm{pop}(t-1)$ is the corresponding value for the previous epoch. A histogram was calculated by plotting $\Re$ values against the corresponding ranks. Initially, such histograms are plotted for each tribe. Once the histograms tend towards $\Re = 0.5$, the population of the tribe consists mostly of non-dominating individuals. The rank ratio is also computed across the tribe boundary in order to check for convergence, and the pareto-front is finally obtained when this inter-tribe histogram also tends towards 0.5.

5. Results and discussion

Here we have developed a PCGA code in C, and used it to calculate the pareto-front consisting of the maximized values of the objective functions shown in Eqs. (1) and (10), subject to the constraint expressed by Eq. (7). All the calculations were performed in a local area network of Pentium machines under a Linux environment.

Due to a high non-linearity associated with the objective functions, a judicious adjustment of the crossover probability was required. Even after that it was not easy to zoom in the pareto-front to the region of equality of both the objective functions. The number of lethals has also gone up linearly with the number of epochs as shown in Fig. 3. In addition, as shown in Fig. 4, both the objective functions tend to spend a considerable number of epochs in the practically non-feasible negative region of the velocity, from which they took a considerable amount of time to recover. Only after running the code for about 5500 epochs the solution converged with the maximized first objective function (u_1) as 0.0976 m/s, closely equal to the second objective function (u_2) as 0.0991 m/s. In general, the second objective function used by us (Eq. (10)) tends to predict casting velocities which are a bit on the higher side. However, at the present state of knowledge no better governing equation is available for the mold oscillation, and the expression used is quite ubiquitous in the continuous casting literature [8]. The pareto solution obtained in this study should be used to determine the upper limit of casting velocity, and considering a 20% confidence limit for such a totally theoretical prediction, the upper limit of billet casting velocity (u_u) can perhaps be fixed at 0.07 m/s. The major billet casters in North America are performing in the casting velocity range 0.010–0.0633 m/s [10], with only a few operating at the upper range. Based upon the present

[2] An epoch in the PCGA strategy can be roughly taken as equivalent to an *iteration* in the derivative-based traditional methods. In any *Evolutionary Computing* scheme, like the one adopted here, a large number of possible solutions are however, present at any epoch or generation. This is in contrast to a single solution that needs to be upgraded during an iteration in most conventional methods.

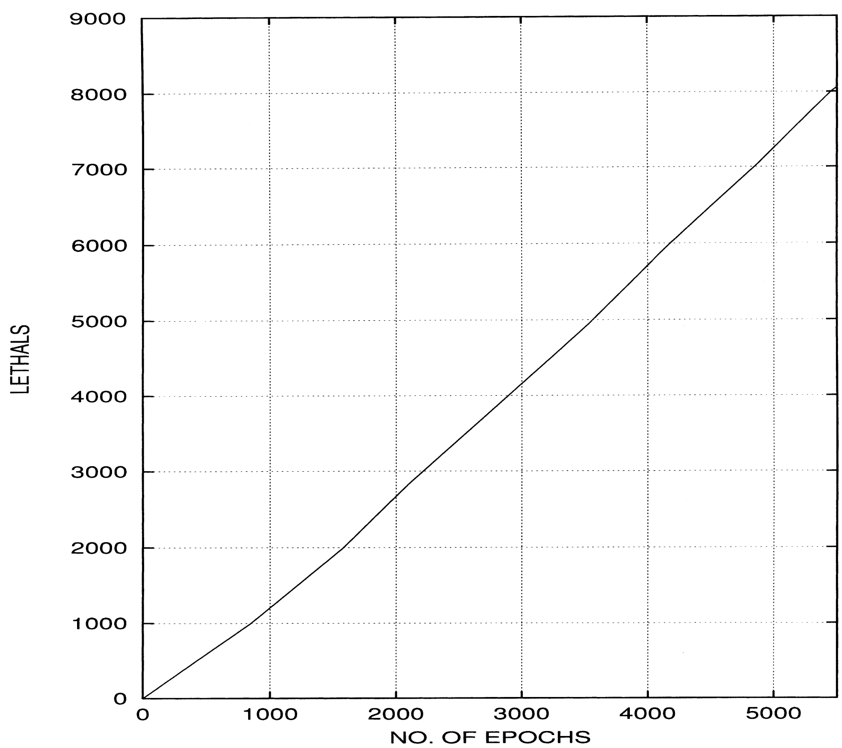

Fig. 3. Linear increase of lethal reproductions with number of epochs.

pareto-optimal solution it appears that with fine tuning of various operational parameters, perhaps still there is some more scope for increasing the casting speed. Also, for a condition similar to what is considered here Geiger and Poirier [14] have theoretically estimated the casting velocity as 0.0508 m/s which is in general agreement with the predictions made here. It is perhaps important to mention at this point that the optimized results would be further accurate if the available heat transfer data were more reliable so that the usage of the correction factors like δ_1 and δ_2 could be avoided. However, at the current level of experimental data availability that appears to be quite impossible. Enough precautions however were taken here so that these correction factors do not lead the solution to a false convergence. The allowable upper and lower limits of these correction factors were kept small, reasonably within the limits of the estimated experimental errors, so that they could only fine tune the optimized result, instead of dominating it.

The optimized values of all the parameters are shown in Table 1. It should be realized at this point that GAs map any variable X from real space to binary utilizing its lower and upper bounds X^{L} and X^{U}, which need to be specified by the user. [3] Thus, during this study, although every parameter shown in Table 1 as taken as a variable, the search for each of them was very much restricted within specified variable bounds provided on the basis of the current industrial practice. This feature of GAs act as a natural safeguard against any spurious solutions which are likely to appear in other techniques where variable bounds need not be specified. For example, it is impossible for a GA-based solution to converge to a shell thickness zero with an infinite casting speed, as any rationally prescribed lower limit on on shell thickness will simply prevent it from

[3] A simple way to achieve that would be a linear mapping, where for a ℓ bit encoding, S the decoded value of the string is related to the variable X as: $X = X^{\mathrm{L}} + S(X^{\mathrm{U}} - X^{\mathrm{L}})/(2^{\ell} - 1)$.

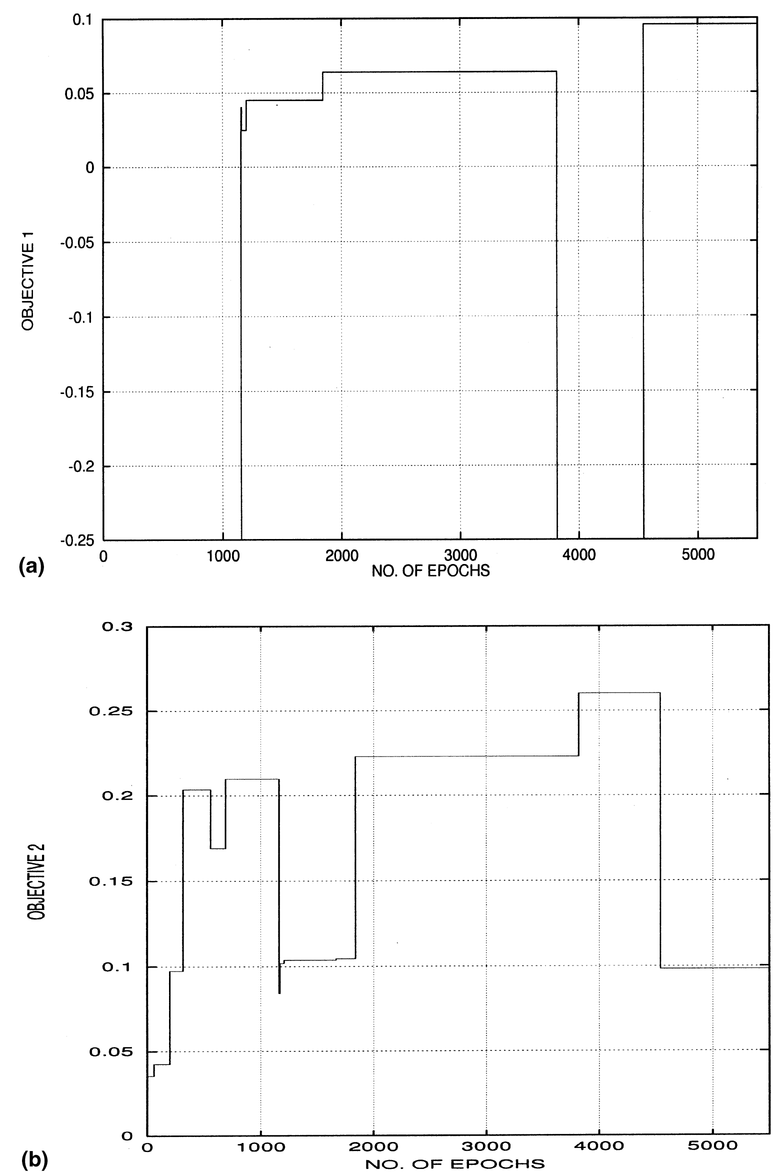

Fig. 4. Typical paths towards convergence. The objective function values are in m/s: (a) for the first objective function; (b) for the second objective function.

becoming vanishingly small. In many other techniques, Simulated Annealing for example, such built in safeguards are non-existing. It was shown in our earlier work [4] that because of this reason Simulated Annealing often predicted practically unachievable casting speeds while genetic algorithms closely followed the industrial trends.

The results obtained by the present optimization scheme seems to be quite reasonable. The present study has indicated a negative strip time of 0.625 s, an oscillation frequency of 3.43 Hz, and a stroke length of 11.76×10^{-3} m, which are perfectly reasonable considering the data available from twenty four North American billet casters [10].

The accuracy with which PCGA is able to predict the casting conditions is therefore quite promising and this premier attempt of applying a pareto-optimal formulation to the continuous

Table 1
Optimized parameters

Variable	Optimized value
T	3.43 Hz
S	11.76×10^{-3} m
t_N	0.625 s
M	0.0111 m
ε	2.53×10^{-1} (−)
δ_1	8.92×10^{-2} kJ m^{-2} s^{-1} K^{-1}
δ_2	-1.69×10^{-3} kJ m^{-2} s^{-1} K^{-1}
T_o	441.88 K
T_S	1450.11 K
T_{Wout}	303.92 K
β	1.65×10^5 kJ m^{-2} s^{-1}
ϕ	7.718 m s^{-1}
u_1	0.0976 m s^{-1}
u_2	0.0991 m s^{-1}
u_u	0.07 m s^{-1}

casting process seems worthy of further exploration. To any researcher in this area PCGA can therefore be recommended as a very powerful modeling tool. It needs to be highlighted at this stage that although no numerical optimization scheme can really guarantee converging to a global optimum, the likelihood of finding it is much higher in a GA or, for that matter, in a PCGA type of computing environment. The reason simply is that the GAs work on the basis of a *population,* which in plain words means that at any point of time, instead of a single guess value or its lone update, we have a large number of possible solutions present in the GA-based techniques, covering the entire search space. PCGA, in addition, uses a multi-population approach (the *tribes* in its terminology) further adding to the population diversity. Thus GA-based solutions are designed to be independent of initial guess values, which are prescribed fully randomly in any good GA code developed to date. The traditional way of altering the initial guess and checking the stability of solution is thus redundant in GAs, where it is usually done by slightly increasing the mutation probability once the solution appears to be stable. If the optimum reached is not global, then one should be able to mutate out of it, thus preventing any premature convergence. All these precautions were taken during this study and we found PCGA to be remarkably steady.

During this study, attention was focused only on the mold region of the caster, and our current efforts are directed towards a more complete analysis of the process, involving its other portions like tundish, spray-cooling region etc. A complete optimization-based model for the continuous casting however warrants many more years of continued research, and in this process the evolutionary computing techniques are likely to contribute in a very big way. Some vital ground work has been done in this study in continuation of our earlier efforts in this area [3,4], which hopefully, will stimulate future GA-based optimization studies of continuous casting in a very near future.

References

[1] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.
[2] M. Mitchell, An Introduction to Genetic Algorithms, Prentice-Hall, New Delhi, India, 1998.
[3] N. Chakraborti, K. Deb, D. Jain, EvoNews, (4 June) (1997) 2.
[4] N. Chakraborti, A. Mukherjee, Ironmaking Steelmaking 27 (2000) 243.
[5] B. Filipic, B. Sarler in: H.J. Zimmermann (Ed.), Proceedings of Sixth European Congress on Intelligent Techniques and Soft Computing (EUFIT '98), vol. 1, Verlag-Mainz, Aachen, Germany.

[6] V. Chankong, Y.Y. Haimes, Multi-Objective Decision Making Theory and Methodology, North-Holland, New York, 1983.
[7] R. Kumar, Ph.D. thesis, University of Sheffield, UK, 1997.
[8] J.K. Brimacombe, I.V. Samarasekera, The Continuous Casting of Steel, Lecture Notes of a course presented at Fluminenese Federal University (UFF), Brazil, 1995.
[9] B.G. Thomas, A. Moitra, D.J. Habing, J.A. Azzi, in: Proceedings of the First European Conference on Continuous Casting, Florence, Italy, September 1991.
[10] I.V. Samarasekera, J.K. Brimacombe, K. Wilder, Iron and Steelmaker 21 (3) (1994) 53.
[11] J. Periaux, M. Sefrioui, B. Mantel, in: D. Quagliarella, J. Periaux, C. Poloni, G. Winter (Eds.), Genetic and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications, Wiley, Chichester, UK, 1998.
[12] S. Obayashi, in: D. Quagliarella, J. Pe'riaux, C. Poloni, G. Winter (Eds.), Genetic and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications, Wiley, Chichester, UK, 1998.
[13] C. Poloni, V. Pediroda, in: D. Quagliarella, J. Pe'riaux, C. Poloni, G. Winter (Eds.), Genetic and Evolution Strategies in Engineering and Computer Science: Recent Advances and Industrial Applications, Wiley, Chichester, UK, 1998.
[14] G.H. Geiger, D.E. Poirier, Transport Phenomena in Metallurgy, Addison-Wesley, Reading, MA, USA, 1973.
[15] J.E. Lait, J.K. Brimacombe, F. Weinberg, Ironmaking Steelmaking 1 (1974) 90.
[16] N. Chakraborti, D. Jain, K. Deb, submitted for publication.
[17] E. Takeuchi, J.K. Brimacombe, Metall. Trans. 15B (1984) 493.
[18] K. Deb, Optimization for Engineering Design: Algorithms and Examples, Prentice-Hall, New Delhi, India, 1995.
[19] C. M. Fonseca, P.J. Fleming, in: S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan-Kaufmann, San Mateo, CA, 1993.
[20] N. Srinivas, K. Deb, Evolutionary Computation 2 (1995) 139.