New Perspective to Continuous Casting of Steel with a Hybrid Evolutionary Multiobjective Algorithm

Karthik Sindhya \textsuperscript{a} & Kaisa Miettinen \textsuperscript{a}

\textsuperscript{a} Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

Available online: 08 Apr 2011

To cite this article: Karthik Sindhya & Kaisa Miettinen (2011): New Perspective to Continuous Casting of Steel with a Hybrid Evolutionary Multiobjective Algorithm, Materials and Manufacturing Processes, 26:3, 481-492

To link to this article: http://dx.doi.org/10.1080/10426914.2010.523913

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
New Perspective to Continuous Casting of Steel with a Hybrid Evolutionary Multiobjective Algorithm

Karthik Sindhya and Kaisa Miettinen

Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland

In this article, we present a new perspective in solving computationally demanding problems such as the optimal control of the continuous casting of steel. We consider a multiobjective formulation of the optimal control of the surface temperature of the steel strand with five objectives, where constraint violations are minimized as objectives because no feasible solutions exist otherwise. A hybrid evolutionary multiobjective algorithm (HNSGA-II) is used to overcome discrepancies of evolutionary multiobjective optimization algorithms such as slow convergence and lack of convergence proof to the Pareto front. HNSGA-II uses NSGA-II as an underlying evolutionary multiobjective algorithm and an achievement scalarization function to scalarize objective functions for local search, which improves the population and speeds up the convergence. This is important because most evolutionary multiobjective algorithms are known to have difficulties with five objective functions. The algorithm is used to generate a set of Pareto solutions showing different trade-offs. However, it is difficult to generate Pareto solutions showing all the different trade-offs with a finite small population size. Hence, we use the preference information related to constraint violations in the weights of the achievement scalarization function to generate solutions with different trade-offs in preferable regions of the Pareto front. In addition, we use clustering and present typical solutions of different clusters so that the most preferred solution to be implemented can easily be identified. We demonstrate the approach and compare the results to previous studies.

Keywords Applications; FEM-based optimization; Nonlinear multiobjective optimization; NSGA-II; Pareto-optimality; Preference information.

INTRODUCTION

Many optimization problems in engineering involve multiple conflicting objective functions, as for example [1–5]. Such problems are called multiobjective optimization problems. Because the objective functions are conflicting, there exist multiple compromise solutions called Pareto (optimal) solutions with different trade-offs (also referred as Pareto front). There are at least two different fields which focus on solving multiobjective optimization problems: multiple criteria decision making (MCDM) [6–9] and evolutionary multiobjective optimization (EMO) [10, 11]. The main difference between the MCDM and EMO fields is that MCDM usually aims at supporting a decision maker (an expert in the problem domain) in finding a single most preferred solution and EMO usually tries to approximate the entire Pareto front with a number of solutions.

EMO algorithms, despite their wide usage, are often criticized for their slow convergence and lack of theoretical convergence proof to the Pareto front. One way to overcome these discrepancies is to utilize scalarized objective functions commonly used in MCDM, which can be subsequently solved using any mathematical programming technique in EMO algorithms. The resulting algorithms are often referred to as hybrid EMO algorithms (or as memetic algorithms). Hybrid EMO algorithms often involve a global phase, which focuses on exploring the entire search space to find promising regions and a local (MCDM-based) phase improves the individuals in a population to a locally optimal solution. Often, EMO algorithms are utilized in the global phase and a local search of a scalarized function using any appropriate mathematical programming technique constitutes a local phase. The main idea here is to enhance convergence speed of the hybrid EMO algorithm in addition to guaranteeing optimality (at least local) of the final population.

Literature has a plethora of hybrid evolutionary multiobjective algorithms. Many of them use a naive neighborhood search procedure (e.g., [12]) or a weighted sum of the objective functions (e.g., [13, 14]). In [15], we used an achievement scalarizing function [16] (described in Section 2) to scalarize the objective functions and subsequently used for local search in our concurrent hybrid NSGA-II [17] approach. For this article, we utilize a new version of a concurrent hybrid NSGA-II algorithm, and we call this algorithm HNSGA-II. The HNSGA-II algorithm addresses two basic issues: choice of individuals for local search and diversity enhancement due to lapse in diversity arising from local search. Here, the population is projected on a hyperplane formed by the worst values of each objective function and clustered. The individuals are subsequently chosen from each of these clusters for local search. In addition, we have incorporated a novel diversity preservation procedure using clustering (described in Section 3) into HNSGA-II.

In this article, we apply the HNSGA-II algorithm to a control problem in the secondary cooling region of continuous casting in steel, described in [18, 19] and references therein. The partial differential equations that represent the cooling process are discretized by the...
finite element and finite difference methods. The resulting objective function to keep the surface temperature of the steel strand near the desired temperature has 325 control variables describing the intensities of water sprays from different sprayers in the secondary cooling region. In addition, the model has four constraints: surface temperature bound constraint, avoiding excessive cooling or reheating on the surface of strand, restricting the length of the liquid pool, and avoiding too low temperatures at the unbending point. It was shown in [19] that this formulation with a single objective and four constraints has an empty feasible region, in other words, the constraints imposed on the cooling process cannot be satisfied simultaneously. Hence, a multiobjective problem (MOP) of minimizing the constraint violations in addition to the existing objective function was formulated in [19]. This MOP with five objective functions was solved in [19] using a scalar version of the interactive NIMBUS method [8, 20, 21]. Later, four objectives of this problem were reconsidered in [22], where the decision maker (DM) was informed about the feasible objective vectors using advanced visualizations of Pareto solutions. More recently in [23], the MOP was successfully solved with a combination of a scalarization-based interactive method and an evolutionary algorithm (used to solve single objective optimization subproblems formulated). There the interactive classification-based NIMBUS method was used to find only such solutions that are desirable to the DM.

The secondary cooling region of the continuous casting of steel has been studied in the literature previously. Among others, minimization of total bulging in the strand using a thermal-bulge model subject to a constraint set on the midwide surface temperature at the exit was studied in [24]. Additionally, water flux was minimized using genetic algorithms, considering spray and radiation cooling regions. Furthermore, the maximization of solidified shell thickness in the mold region using genetic algorithms was studied in [25]. On the other hand, multiobjective optimization using genetic algorithms has been applied previously in different phases of the continuous casting process. In [26], a Pareto-converging genetic algorithm [27] was used to minimize casting velocity in terms of mold oscillation in the mold region. The maximization of casting production rate as a function of casting constraints was carried out in [28]. There, the decision rules, the weighting method and a genetic algorithm together with a numerical heat transfer model were used to simulate operating conditions for a continuous casting machine. Furthermore, in [29] the mold parameters were optimized to maximize casting velocity using genetic algorithms and other techniques. In addition, the cooling conditions were optimized in [30, 31] by employing a knowledge base of operational parameters, a genetic algorithm and weighing method. More recently in [32], an evolutionary neural network approach (PPNNGA) was used to pick the optimum network model for representing the objective functions as a metamodel. There, a simultaneous maximization of casting velocity and the shell thickness at mold exit and minimization of reheating at the spray-radiation cooling interface along with the simultaneous maximization of casting velocity were used as objective functions. Subsequently, this multiobjective problem was optimized using a multiobjective predator-prey genetic algorithm [33]. A comprehensive literature review on the applications of EMO algorithms in materials science and engineering is presented in [34].

The purpose of this article is to provide a new perspective in solving computationally demanding problems like the one related to the optimal control of the surface temperature in the secondary cooling region of the continuous caster by using evolutionary multiobjective algorithms. As described earlier, evolutionary multiobjective algorithms have discrepancies regarding lack of convergence proof and slow convergence. Hence, in this article we use HNSGA-II to overcome such discrepancies. Even though the continuous casting of steel has been studied previously in the literature, no studies consider an evolutionary multiobjective algorithm to find out different trade-offs that may exist in a MOP with as many as five objectives. Furthermore, the weights of the achievement scalarizing function (used as a scalarizing function in the HNSGA-II algorithm) can be embedded with preference information of the DM. With the achievement scalarizing function members of an EMO population can be projected in preferable directions to obtain solutions with more preferable trade-offs. This is especially useful in MOPs with more than three objectives, as it is very difficult to approximate the entire Pareto front with a small finite population size. Finally, we utilize clustering to represent the variety of trade-offs to the DM without introducing too much cognitive load on him/her because comparison of solutions would otherwise be very demanding if not impossible. By studying the clustered solutions, the DM can gain valuable insight into the problem.

The rest of the article is organized as follows. We introduce the concepts of multiobjective optimization and achievement scalarizing function in Section 2. In Section 3, a brief description of the HNSGA-II algorithm is provided. The problem formulation related to the secondary cooling region in continuous casting of steel is described in Section 4, and the numerical results of applying HNSGA-II showing different existing trade-offs are shown in Section 5. Finally, we conclude in Section 6.

**Some concepts**

In this section, some concepts relevant to this article regarding multiobjective optimization are first described. Subsequently, we introduce an achievement function used for the local phase in the solution approach used.

In this article, we deal with a multiobjective optimization problem of the form

\[
\text{minimize } \{ J_1(\mathbf{u}), J_2(\mathbf{u}), \ldots, J_k(\mathbf{u}) \}
\]

subject to \( \mathbf{u} \in U \subset \mathbb{R}^n \),

with \( k \geq 2 \) conflicting objective functions \( J : \mathbb{R}^n \to \mathbb{R} \). We denote the vector of objective function values by \( \mathbf{J}(\mathbf{u}) = (J_1(\mathbf{u}), J_2(\mathbf{u}), \ldots, J_k(\mathbf{u}))^T \) to be called an objective vector. The decision (or control) vectors \( \mathbf{u} = (u_1, u_2, \ldots, u_n)^T \) belong to the feasible region \( U \).

The basic definitions of optimality are the following ones.
DEFINITION 1. A decision vector $u^* \in U$ for problem (1) is a Pareto (optimal) solution, if there does not exist another $u \in U$ such that $J_i(u) \leq J_i(u^*)$ for all $i = 1, 2, \ldots, k$ and $J_j(u) < J_j(u^*)$ for at least one index $j$. An objective vector is Pareto optimal if the corresponding decision vector is Pareto optimal. This definition also dictates global Pareto optimality.

DEFINITION 2. Let $B(u^*, \delta)$ denote an open ball with a center $u^*$ and a radius $\delta > 0$, $B(u^*, \delta) = \{u \in \mathbb{R}^n \mid \|u - u^*\| < \delta\}$. A decision vector $u^* \in U$ is said to be locally Pareto optimal if there exists $\delta > 0$ such that $u^*$ is Pareto optimal in $U \cap B(u^*, \delta)$. An objective vector is locally Pareto optimal if the decision vector corresponding to it is locally Pareto optimal.

In general, problem (1) typically has many Pareto solutions. The set of such Pareto solutions in the objective space is referred as a Pareto optimal set or Pareto front.

Since problem (1) involves more than one objective, scalarization techniques are commonly used in the MCDM field to obtain Pareto solutions. Literature has a plethora of scalarization techniques [8], and among them an achievement scalarizing function [8, 16] is commonly used. It was proposed in [16], and it uses a reference point $\bar{z} \in \mathbb{R}^k$. In the context of this article, reference points are objective vectors in the population chosen for local search.

An example of an achievement scalarizing function [8, 16] is given by

$$\text{minimize} \quad \max_{i=1}^{k} [w_i(J_i(u) - \bar{z}_i)] + \rho \sum_{i=1}^{k} [w_i(J_i(u) - \bar{z}_i)],$$

subject to $u \in U$, (2)

where $w_i = \frac{1}{z_{i_{\text{max}}} - z_{i_{\text{min}}}}$ is a weight factor (usually used for normalizing) assigned to each objective function $J_i$. Here we utilize in these factors maximum and minimum values of each objective in a generation represented as $z_{i_{\text{max}}}$ and $z_{i_{\text{min}}}$, respectively. An augmentation coefficient ($\rho$) takes a small positive value. The above problem produces (properly) Pareto optimal solutions with bounded trade-offs. The augmentation terms may also improve the computational efficiency (resulting with a shorter computation time) for the above problem, see [35].

The advantages of using an achievement scalarizing function are the following ones, according to [8]:

1. The optimal solution of an achievement scalarizing function is always Pareto optimal;
2. Any (properly) Pareto optimal solution can be obtained by changing the reference point;
3. The optimal value of an achievement scalarizing function is zero, when the reference point is Pareto optimal.

In the next section, we introduce briefly the HNSGA-II algorithm utilizing the achievement scalarizing function.

HNSGA-II ALGORITHM

The hybrid NSGA-II (HNSGA-II) algorithm is a concurrent hybrid algorithm. The HNSGA-II algorithm basically differs from other hybrid algorithms and the original NSGA-II algorithm in three aspects:

1. A better choice of individuals for local search: Individuals in a population are projected on a hyperplane formed by the worst values of each objective function and clustered. In this algorithm, we use the $k$-means clustering algorithm [36] to cluster the population. An individual from each cluster is improved by solving an achievement scalarizing function with an appropriate (local) optimizer to be able to project it in different parts of the Pareto front. This will enhance the diversity of the population.

2. An enhanced diversity preservation module: When a local search is incorporated in a concurrent mode, it is likely to create an individual far from the current population. Such an individual is generated due to the single objective optimization of an achievement scalarizing function. We call such individuals as super individuals (solution $S$ in Fig. 1). Due to their increased selection pressure, they can cause a rapid deletion of dominated solutions leading to loss of diversity in the population. In such a situation, we apply diversity enhancement. Figure 1 shows the diversity enhancement of HNSGA-II. The clustered subpopulations $c_1$, $c_2$, and $c_3$ as shown in Fig. 1 are subjected to nondominated sorting [11] to form different nondomination levels. A new population for the next generation is generated by choosing individuals first from the best front of each cluster and then successively until the last front, till the population size $N$ is achieved as shown in Fig. 1.

![Figure 1.—Diversity enhancement in HNSGA-II.](image-url)
(3) **Diversity check:** When the population is clustered as described in item (1) above, a cluster quality index is calculated. The cluster quality index is defined as

\[ Q_{gen} = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{|RP_i|} \sum_{C_j \in RP_i} (D(C_j, \sigma_i)), \]

where \( K \) is the number of clusters, \( \sigma_i \) represents the cluster centroid of cluster \( i \) and \( C_j \) represents a point in the cluster \( i \). \( D(C_j, \sigma_i) \) is the distance of the point in cluster \( i \) to its centroid and \( |RP_i| \) is the number of points in the cluster \( i \). The cluster quality index \( Q_{gen} \) at generation \( gen \) is used as a lower bound \( Q_{low} \) for the next \( g \) generations. If \( Q_{gen} \) at any \( g \)th generation falls below \( Q_{low} \), the population is declared to have a bad diversity and otherwise it is declared to have a good diversity. A good diversity is essential for the population to explore the search space and to finally yield a population representing the entire Pareto front. In what follows, we describe how good or bad diversity is treated in the algorithm.

The HNSGA-II algorithm consists of the following steps:

(1) Generate a random initial population \( P_0 \) of size \( N \) and set generation count \( t = 0 \).

(2) Sort population to different nondomination levels (fronts) \([11]\), and assign each solution a fitness equal to its nondomination level (1 is the best level).

(3) Create offspring population \( Q_t \) of size \( N \) using, e.g., binary tournament selection, SBX \([37]\) recombination and polynomial mutation \([11]\) operations.

(4) Combine the parent and the offspring populations and create \( R_t = P_t \cup Q_t \).

(5) Perform nondominated sorting to \( R_t \), and identify different fronts \( F_i \), \( i = 1, 2, \ldots \) etc.

(6) Set new population \( P_{t+1} = \phi \). Set a count \( i = 1 \) and as long as \( |P_{t+1}| + |F_i| \leq N \), perform \( P_{t+1} = P_{t+1} \cup F_i \) and \( i = i + 1 \). Here, \( |P_{t+1}| \) and \( |F_i| \) are the numbers of individuals in the population \( P_{t+1} \) and the front \( F_i \), respectively.

(7) Perform the crowding-sort procedure \([11]\) and include the most widely spread \((N - |P_{t+1}|)\) members of \( F_i \) by using the crowding distance values in the sorted \( F_i \) to \( P_{t+1} \).

(8) Project the population \( P_{t+1} \) onto a hyperplane formed by the worst values of each objective function to get the projected population \( P_{t+1} \) and then cluster \( P_{t+1} \) into \( k + 1 \) clusters.

(9) Tag the cluster number of the individuals in the population \( P_{t+1} \) to the corresponding individual in the population \( P_{t+1} \).

(10) If the population \( P_{t+1} \) has a good diversity, goto Step 11, otherwise go to Step 12.

(11) The population \( P_{t+1} \) is subjected to local search by picking individuals with a probability \( P_{local} \) from different clusters.

(12) If the population \( P_{t+1} \) has a bad diversity, send the population \( R_t \) to the enhanced diversity preservation module (described earlier in this Section).

(13) Check if the termination criterion is satisfied (in this article, we consider a fixed budget of function evaluations as a termination criterion). If yes, go to step 14, else set \( t = t + 1 \), and return to step 2.

(14) Perform local search on all the individuals in the population \( P_{t+1} \) to get the final population.

In the above algorithm, the convergence speed is enhanced due to the local search in Step 11. To preserve diversity in the population, the local search in Step 11 is only performed when there is good diversity in the population. All individuals in the population \( P_{t+1} \) are subjected to local search in Step 14 to guarantee (at least local) optimality of the final population. In addition, the algorithm aims at enhancing convergence in Step 12.

In the next section, we briefly describe the continuous casting process and then present our MOP.

**Problem Description**

Continuous casting is a process in which molten metal is solidified into semi finished slabs. Figure 2 shows an overview of a continuous casting machine. Molten steel is fed from a tundish into a water cooled mold (primary cooling region). In this mold, the steel strand obtains a solid shell. The strand exits at the base of the mold \( z_1 \) and is then supported by rollers and cooled by water sprays in the secondary cooling region. Here, more heat is extracted so that near the exit of the spray cooling region the final solidification of the strand is complete. The secondary cooling region is further subdivided into a number of cooling zones, each comprising of a group of spray nozzles. After the secondary cooling region (at point

![Figure 2.—Overview of the continuous casting process.](image-url)
Mathematical Model

The mathematical model used here is given in more detail in [19] with further references. The model consists of a two dimensional heat conduction equation, and we make the following assumptions:

(1) Due to the high withdrawal rate and low thermal conductivity of steel, the withdrawal direction is neglected;

(2) The casting speed \( v \) is constant.

In our study, the casting process between \( z_4 \) and \( z_5 \) is considered and a time \( t_i = z_i/v \) for \( i = 1, \ldots, 4 \) is defined. The time \( t_i \) is the time when the cross-section of the strand crosses the point \( z_i \) with a constant speed \( v \).

The temperature distribution is denoted by \( y = y(x,t) \), where \( x \) belongs to the two-dimensional cross-section of the strand and \( t \in [t_1, t_4] \). This temperature distribution is calculated by solving an initial-boundary value problem as a system of equations consisting of enthalpy function \( H \) and the Kirchhoff's transformation \( K \). In the initial-boundary value problem, the temperature dependent quantities, steel density \( \rho \), specific heat \( c \) and thermal conductivity \( k \) are defined separately in several disjoint intervals and inside each interval they are assumed to be constants. Furthermore, the zone between the solidus \( y_s \) and the liqudus \( y_l \) temperatures is regarded as mushy. The temperature distribution at the end of the mold (denoted by \( y_m \)) is assumed to be known and is also considered as the initial condition.

The finite element method (FEM) is used to discretize the system in space and the finite difference method (FDM) is used to discretize the system with respect to time. We consider a regular triangulation, with a maximum length of the triangle, \( d \) (grid parameter), \( x_i, i = 1, \ldots, N \) denote the nodes of the triangulation and \( x_i, i = 1, \ldots, P \) denote the nodes on the boundary. Here piecewise linear basis functions are used. The time interval is divided into subintervals with the points \( t_1 = t^{k_1} < \ldots < t^{k_{i-1}} < t^{k_i} = t_4 \), such that time events \( t_2 \) and \( t_3 \) are also among the points (i.e., \( t_i = t^{k_i} \) with some \( k_i \)). Additionally, we define \( \Delta t^k = t_i - t_{i-1} \).

A vector \( y^k = (y^k_1, \ldots, y^k_N) \) is used to approximate the temperature \( y \), where \( y^k_i \) denotes the approximate value of \( y(x_i, t^k) \). Furthermore, the heat transfer coefficient \( u \), the enthalpy function \( H \), and the Kirchhoff's transformation \( K \) are approximated similarly by the vectors \( u^k \), \( H(y^k) \) and \( K(y^k) \), respectively. Additionally, the nodal values of the water spray temperature \( y_{wat} \), the exterior temperature \( y_{ext} \), and \( y_i \) are assumed to be contained in the vectors \( y^k_{wat}, y^k_{ext}, y^k_i \), respectively.

Utilizing the fully implicit scheme in approximating the time derivative \( H \), we obtain the following system of algebraic equations

\[
\begin{align*}
    y^{k+1}_1 &= y^k_1, \\
    \tilde{M}^k H(y^k) + AK(y^k) + \tilde{B}(u^k)y^k + \sigma \epsilon B[y^k]^4 &= \tilde{M}^k H(y^{k-1}) + \tilde{B}(u^k)y^k_{wat} + \sigma \epsilon B[y^k_{ext}]^4, \\
    K_1 < k &< K_2, \\
    \tilde{M}^k H(y^k) + AK(y^k) + \sigma \epsilon B[y^k_{ext}]^4 &= \tilde{M}^k H(y^{k-1}) + \sigma \epsilon B[y^k]^4, \\
    K_2 < k &< K_4,
\end{align*}
\]

where \( \sigma \) is the Stefan–Boltzmann constant, \( \epsilon \) is the emissivity factor, \( M \) and \( A \) are the mass and stiffness matrices, respectively, \( \tilde{M}^k = M/\Delta t^k \), and \( B(u) \) is the Kirchhoff matrices obtained by using the trapezoidal rule in approximating the boundary integrals. It has been shown in [38] that the system (3) has a unique and locally Lipschitz continuous solution \( u \mapsto y(u) \). As \( H \) and \( K \) are not differentiable everywhere, the mapping \( u \mapsto y(u) \) is nondifferentiable. The above system is solved by using modified successive overrelaxation at the interior nodes and the Newton Raphson method on the boundary nodes.

Optimization Problem

The optimization problem that we are interested in here is a temperature control problem in the secondary cooling region to obtain steel of good quality. The problem is to minimize the variation of the surface temperature distribution on the boundary of the steel strand to be close to a pre-defined surface temperature distribution \( y_d = y_d(x, t) \). In short, our aim here is to minimize

\[
J_1(u, y) = 0.5 \sum_{k=K_1+1}^{K_4} \Delta t^k (y^k - y^k_d)^T B(y^k - y^k_d),
\]

where \( y^k_d \) contains the nodal values of \( y_d \).

In addition to the above objective function, the optimization problem has four state constraints:

(1) Minimum surface temperature: The surface temperature in the secondary cooling region is bound to prevent the...
formation of cracks and shape defects in the products. Therefore, we define the upper and lower bounds \( y_{\text{max}}^k \) and \( y_{\text{min}}^k \) in that region, respectively,

\[
y_{\text{min}}^k \leq y_i^k \leq y_{\text{max}}^k, \quad K_1 < k \leq K_4, \quad 1 \leq i \leq P. \tag{5}
\]

(2) **Reheating and cooling rate:** As the strand passes from zone to zone, the reheating or excessive cooling of the surface may be caused. The re-heating of the surface generates tensile stresses near the solidification front and causes internal cracks. Rapid cooling generates tensile stresses near the surface and can enhance or create new cracks. Thus, we define the following constraint to prevent reheating and rapid cooling:

\[
y_{\text{min}}^k \leq \frac{y_i^k - y_{i-1}^k}{\Delta t_i^k} \leq y_{\text{max}}^k, \quad K_1 < k \leq K_4, \quad 1 \leq i \leq P, \tag{6}
\]

where \( y_{\text{min}}^k \) and \( y_{\text{max}}^k \) are appropriate constants.

(3) **Length of the liquid pool:** The length of the liquid pool is restricted by quality requirements and safety reasons. Thus, the temperature of the strand should be lower than the solidus temperature \( y_S \) after the point \( z_3 \), that is,

\[
y_i^k \leq y_S, \quad K_3 < k \leq K_4, \quad 1 \leq i \leq N. \tag{7}
\]

(4) **Surface temperature at the unbending point:** Unbending of the strand causes tensile stresses on the upper surface and compressive stresses on the lower surface. To avoid defects resulting from unbending of the strand, the temperature at the unbending point is maintained outside the low ductility range of the steel. Thus, we choose a minimum value for the temperature \( y_{\text{duc}} \) at \( z_4 \), and we have the following constraint:

\[
y_{\text{duc}} \leq y_i^{K_4}, \quad 1 \leq i \leq N. \tag{8}
\]

The above state constraints are highly conflicting and form an empty feasible region as pointed out in [19]. Hence, it is not possible to solve the above optimization problem by traditional means (i.e., as a single objective optimization problem). As suggested in [19], we reformulate the problem and convert the state constraints as four more objectives. Next, we present them in a discrete form. We follow [19] and handle these constraints by using a penalization technique with exact penalty functions. We consider sets \( C_i, \quad i = 2, \ldots, 5 \) of \( y \) satisfying the constraints (5)-(8) and denote by \( P_i, \quad i = 2, \ldots, 5 \), the projection operators (component-wise projections) into the sets \( C_i \), respectively. We get four more objectives:

(1) **Minimum surface temperature:**

\[
J_2(u, y) = \left[ \sum_{k=K_1+1}^{K_4} \Delta t_i^k (p_i^k)^T B p_i^k \right]^{\frac{1}{2}}, \quad \text{where} \quad p_i^k = y_i^k - P_2 y_i^k, \quad K_1 < k \leq K_4. \tag{9}
\]

(2) **Reheating and cooling rate:**

\[
J_3(u, y) = \left[ \sum_{k=K_1+1}^{K_4} \Delta t_i^k (p_i^k)^T B p_i^k \right]^{\frac{1}{2}}, \quad \text{where} \quad p_i^k = y_i^k - y_{i-1}^k, \quad K_1 < k < K_4. \tag{10}
\]

(3) **Length of the liquid pool:**

\[
J_4(u, y) = \left[ \sum_{k=K_1+1}^{K_4} \Delta t_i^k (p_i^k)^T B p_i^k \right]^{\frac{1}{2}}, \quad \text{where} \quad p_i^k = y_i^k - y_{i-1}^k, \quad K_1 < k < K_4. \tag{11}
\]

(4) **Surface temperature at the unbending point:**

\[
J_5(u, y) = \left[ (p_3)^T M p_3 \right]^{\frac{1}{2}}, \quad \text{where} \quad p_3 = y_i^{K_4} - P_5 y_i^{K_4}. \tag{12}
\]

In addition to the objectives (4) and (9)-(12) to be minimized, for every cooling region an upper bound for the spray water flow rate is fixed. The control \( u \) includes the effect of the supporting rollers and \( u \) has a strictly positive value even if the water sprays are switched off. Thus, we have box constraints

\[
u_{\text{min}}^k \leq u_i^k \leq u_{\text{max}}^k, \quad K_1 < k < K_2, \quad 1 \leq i \leq N. \tag{13}
\]

where \( u_{\text{min}}^k \) and \( u_{\text{max}}^k \) are constants. These box constraints (13) are augmented by a linear inequality for each cooling zone, which describes the upper bound for the values of \( u \) in each zone. The feasible region \( U \) is defined by these constraints. After the discretization, the dimension of the state \( y \) is 1666 and the dimension of the control variable \( u \) is 325. For details, see [19] and references therein.

Finally, our MOP is of the form

\[
\begin{align*}
\text{minimize} & \quad \{ J_1(u, y), J_2(u, y), \ldots, J_5(u, y) \} \\
\text{subject to} & \quad \text{state equation (3) and control constraints (13).}
\end{align*}
\]

Formulation (14) enables solving the problem which otherwise has no solutions. Because of the special nature of objectives \( J_2 - J_5 \), minimizing their values, that is minimizing the constraint violations of the state constraints is very important. This means that minimizing the value of \( J_1 \) is not equally important. Besides, from a practical point of view, as it is very difficult to determine an optimal temperature distribution \( y_d \) leading to steel of good quality.

**Numerical experiments**

In this section we present the results obtained by using our HNSGA-II algorithm for solving problem (14).
The optimizer used for local search in the HNSGA-II algorithm is the proximal bundle method described in [39]. We use this non-differentiable solver to minimize the achievement scalarizing function because the problem has non-differentiable functions. The HNSGA-II algorithm has been implemented in the C language and is coupled with the mathematical model of the continuous casting process implemented in Fortran 77.

For the sake of convenience, we next briefly recall the physical meanings of the objective functions:

(a) $J_1$—keep the surface temperature in the secondary cooling region near the desired temperature;
(b) $J_2$—keep the temperature between the upper and lower bounds;
(c) $J_3$—avoid excessive cooling and reheating of the surface;
(d) $J_4$—restrict the length of the liquid pool; and
(e) $J_5$—avoid too low temperatures at the unbending point.

As discussed earlier, the last four objectives ($J_2 - J_5$) are considered to be more important than the first one, because we want the constraints to be satisfied as well as possible. This information is regarded as the preference information of the DM. Additionally, as the last four objectives originally represent constraints (henceforth, in this section we refer to them as constraints), we prefer their values to be as close to zero as possible.

HNSGA-II involves six parameters and their values are set as following:

(1) Population size ($N$): 200;
(2) Crossover probability ($P_c$): 0.9;
(3) Mutation probability ($P_m$): 1/(number of variables);
(4) SBX distribution index ($\eta_c$): 10;
(5) Mutation distribution index ($\eta_m$): 50;
(6) Cluster quality index update: every 10 generations.

The parameters settings for $N$, $P_c$, and $P_m$ are commonly set as above in the literature. The setting for $\eta_c$ is used as 10 to produce offsprings far from the parents, so that a better global search of the search space can be achieved, and the value for $\eta_m$ is set as 50 so that only a small perturbation in the vicinity of the chosen individual for mutation is achieved. The cluster quality index update is set for every 10 iterations based on heuristics. The proximal bundle method involves an accuracy parameter $\epsilon$ used in the stopping criterion, which is set to $10^{-4}$. A budget of 10,000 function evaluations was pre-fixed and the resulting nondominated solutions were subjected to local search to guarantee (at least local) optimality. As described in [40], a reference point can be projected in a preferable direction to the DM by altering the weights of the achievement scalarizing function. Here, we altered the weight for the objective function $J_1$ to be 0.1, so that minimizing constraint violations is slightly preferred over the objective function $J_i$. The solutions of the final population were clustered into 10 clusters, to identify different trade-off. It must be noted that the number of clusters was fixed arbitrarily as it is impossible to know a priori the number of clusters. In our case, finding the optimal number of clusters is not relevant because we only want to show different trade-offs families (represented by clusters) to the DM. If so desired, the number of clusters can be changed according to the wish of the DM and the final population reclustered. A value path plot [8, 41] is
used to visualize solutions (i.e., five dimensional objective vectors) in different clusters. The horizontal axis of the value path plot contains the objective functions, and the vertical axis contains the corresponding objective function values. The range of each of the objective functions can be easily visualized in a value path plot. The value path plots for the ten clusters are shown in Figs. 3–12. The role of these plots is to show to the DM how similar or different solutions there are in each cluster and demonstrate typical trade-offs in each cluster. This information gives to the DM some insight about which of the original constraints can be satisfied at what price in the others. One can see that many of the clusters are not interesting as one or two original constraints can be satisfied.

However, next we pick one representative solution (arbitrarily) from each cluster and plot a value path plot (Fig. 13), to get an overall idea of the trade-offs that a DM can expect between different objectives. We can observe the following trends in Fig. 13:

(1) Constraint $J_2$ is zero in all solutions, indicating it is easy to keep the surface temperature of the strand between the upper and lower bounds;

(2) Constraints $J_2$ and $J_4$ are comparatively easier to be satisfied, when compared to constraints $J_3$ and $J_5$;

(3) If the constraint violations are low for all constraints, the value of the objective function $J_1$ is high. It can be
observed that for at least two representative solutions the value of the first objective function $J_1$ is greater than 1 and their constraint violations are low;

(4) As expected, there are no solutions where all four constraints are satisfied;

(5) There exists one solution for which three constraints ($J_3$, $J_4$, and $J_5$) are nearly satisfied. From the perspective of the DM, this solution is better than the other solutions in Fig. 13.

The different Pareto optimal solutions provide a wealth of information to the DM operating the continuous caster, which can be summarized as follows:

(1) It is not possible to maintain the surface temperature distribution of the strand as close to the predefined temperature distribution simultaneously satisfying all the state constraints (Figs. 3–12).

(2) It is possible to maintain the surface temperature in the secondary cooling region between the upper and lower bounds (mentioned in Section 4), but one or more of the other state constraints $J_3$, $J_4$, and $J_5$ will be violated (Figs. 3–12).

(3) The temperature of strand at the unbending point can be maintained above or around the low ductility range of steel only by allowing the temperature of strand above the solidus temperature after the point $z_3$ and with violations in the rapid cooling and reheating of the surface of the strand. In other words, the strand with low constraint violations in $J_5$ may not be solidified before the unbending point and additionally may be subjected to reheating or rapid cooling when passed between different zones [Figs. 3, 4, and 10].

(4) If the solidification of the strand is complete before the unbending point, the temperature at the unbending point cannot be maintained above the low ductility range of steel. Additionally, the strand may be subjected to reheating or rapid cooling when passed between different zones (Figs. 5, 9, 11, and 12).

(5) It is possible to maintain the surface temperature distribution of the steel close to the predefined temperature distribution, if the strand is solidified before the unbending point. But the strand may have to undergo reheating or rapid cooling when passed between different zones and have a lower temperature than the low ductility range of steel at the unbending point [Figs. 9 and 11].

(6) Finally, the strand does not undergo excessive reheating and rapid cooling on the surface and is solidified before the unbending point (in addition to the $J_5$ constraint as
always being satisfied) by allowing a large deviation in the surface temperature distribution of steel from the predefined temperature distribution and a lower temperature of strand than the low ductility range of steel at the unbending point (Fig. 8).

The solution with the maximum number of constraints satisfied belongs to the cluster 6 (Fig. 8). After having studied the clusters, Figs. 3–12, the DM can choose from the cluster 8, the best solution that matches his/her preferences, which can be for example $J_1, J_2, J_3, J_4, J_5 = 1.4287, 0.0, 0.0005, 0.00001, 0.57$.

Now, having studied the different trade-offs, let us again consider cluster 6. Here we can see that by allowing the objective function value of $J_1$ to impair, we can better satisfy the constraints. Hence, we can further make a small change in the algorithm to incorporate this information. For the local search in the HNSGA-II algorithm, we further modify the weight corresponding to the objective function $J_1$ as 0.001. This modified weight is only used with a small probability of 0.1. This modification allows the HNSGA-II algorithm to explore the preferable region of lower constraint violations with a small probability. Thus, we expect in addition to different trade-offs as seen before, to obtain Pareto solutions which better satisfy the preferences of the DM, if such solutions exist. All other parameter for HNSGA-II remain the same. Again, the solutions obtained are clustered into ten clusters.

The value path plot of the representative solutions from each cluster is shown in Fig. 14 (we do not show the clusters here to save space). It can be observed from Fig. 14 that there exist different solutions where the objective function $J_1$ is further impaired and the constraint violations in $J_2$ has further decreased in addition to constraints $J_3, J_4$, and $J_5$ being fully satisfied. The solutions showing this trend in Fig. 14 belong to the cluster 10. In Fig. 15, we show the different solutions present in cluster 10. The DM has a number of solutions in cluster 10, all of which have very low constraint violations and can choose the best solution he/she desires. Let us note that by studying clusters, the DM does not need to analyze large amounts of data and can affect the population generated by one’s preference information if so desired, as was done here. In the literature [19], the best solution suggested using an interactive method NIMBUS is $J_1, J_2, J_3, J_4, J_5 = 2.12, 0.0, 0.0, 0.0, 0.21$, which is in fact comparable to one of the solutions present in the cluster 10, i.e., $J_1, J_2, J_3, J_4, J_5 = 2.05, 0.00, 0.00, 0.00, 0.23$.

Furthermore, the solution approaches of the HNSGA-II algorithm and an iterative interactive method NIMBUS
are different. Hence we cannot directly compare the final solutions only.

**Conclusion**

We have shown that even a computationally demanding MOP like the continuous casting of steel can be efficiently solved with a population based solver if an appropriate solver like the HNSGA-II algorithm is used. To gain fast and accurate convergence, we have used an augmented achievement scalarization function as a scalarizing function during local search which is a part of the HNSGA-II algorithm. In addition, preference information was also easily embedded in the weights of the achievement scalarization function, to achieve desirable trade-offs to the DM. The enhanced diversity preservation techniques has helped the HNSGA-II to generate a diverse set of solutions. All solutions obtained are Pareto optimal thanks to applying local search at the end of the solution process. Finally, we cluster the final population. In this way of solving a multiobjective optimization problem, we support the DM by providing him/her knowledge about the different trade-off solutions that exist and he/she can then choose the most preferred solution as the final one even in the case of five objectives.

This study has demonstrated how a hybrid EMO algorithm (HNSGA-II) can be used to solve an optimal control problem related to the secondary cooling zone in the continuous casting of steel. The objective here was to produce steel of good quality by maintaining the surface temperature of the strand close to the desired temperature, without violating the technological constraints. The constraints were highly conflicting and formed originally an empty feasible set. Hence, a multiobjective problem was formulated by considering the constraint violations as additional objectives, and we could find a set of Pareto solutions with different trade-offs, provide insight of the problem, and eventually support the DM in finding a satisfying solution. Besides demonstrating the potential of efficient evolutionary algorithms in handling complicated problems, this case demonstrates a situation how one can end up in having to solve a multiobjective optimization problem, even though the original formulation had only one objective.

**Acknowledgments**

The authors would like to warmly thank Mr. Vesa Ojalehto for his valuable help in connecting the problem model with the optimization software used. Thanks to Dr. T. Männikkö for the secondary cooling model and Fig. 2.

**References**


