



Bainite transformation time model optimization for Austempered Ductile Iron with the use of heuristic algorithms

Izabela Olejarczyk-Woźńska¹ , Andrzej Opaliński¹ , Barbara Mrzygłód¹ ,
Krzysztof Regulski^{1*} , Wojciech Kurowski

¹ AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Krakow, Poland.

Abstract

The paper presents the application of heuristic optimization methods in identifying the parameters of a model for bainite transformation time in ADI (Austempered Ductile Iron). Two algorithms were selected for parameter optimization – Particle Swarm Optimization and Evolutionary Optimization Algorithm. The assumption of the optimization process was to obtain the smallest normalized mean square error (objective function) between the time calculated on the basis of the identified parameters and the time derived from the experiment. As part of the research, an analysis was also made in terms of the effectiveness of selected methods, and the best optimization strategies for the problem to be solved were selected on their basis.

Keywords: heuristic optimization, bainite, ADI, Particle Swarm Optimization, Evolutionary Optimization algorithm

1. Introduction

Optimization is a field of science whose goal is to find the best (minimum or maximum) solution to a given problem that meets all constraints and conditions, and that can be presented in the form of a mathematical model. There are many optimization methods that can be selected individually depending on the problem under consideration. In the case of the optimization of the parameters of models for which the probability of the occurrence of a significant number of local minima is high, and the classical methods are too slow, heuristic search methods are usually used (Gili et al., 2019).

The most important features of a heuristic search method include:

- improving efficiency in solving complex problems;
- finding a solution along the most likely path while avoiding less promising paths;
- avoiding checking so-called “dead ends”, for example, by using previously acquired information;
- formulation of simple criteria for selecting directions of conduct without unequivocally defining good and bad states (Boccardo et al., 2015; Hepp et al., 2012; Zimba et al., 1999).

The use of optimization algorithms to adapt numerical models to experimental values is widely used in the manufacturing industry (Essaid et al., 2018; Tang et al., 2013). Their use contributes to reducing production costs, for example by reducing the number of laboratory tests. However, the application of optimization in the

*Corresponding author: regulski@agh.edu.pl

ORCID ID's: 0000-0001-9838-4791 (I. Olejarczyk-Woźńska), 0000-0002-9730-9594 (A. Opaliński), 0000-0001-9619-5305 (B. Mrzygłód), 0000-0001-8080-2254 (K. Regulski)

© 2022 Authors. This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License requiring that the original work has been properly cited.

metal casting sector is significantly limited. In the case of ADI manufacturing, one can only find the results of studies from a few research centers in the literature, where authors have attempted to model changes in the microstructure of ductile iron in order to obtain ADI (Boccardo et al., 2015; Hepp et al., 2012; Zimba et al., 1999). The model developed by Zimba (Zimba et al., 1999) concerned the austenitic transformation of cast iron with a ferritic matrix. This model makes it possible to calculate changes in the carbon concentration in austenite over time, depending on the size of the so-called ferritic cells of ductile iron. It was assumed that each cell covers one graphite sphere and that the cell boundary coincides with the ferrite grain boundary. In turn, the modeling of the austenite \rightarrow ferrite transformation was tackled, among others, by Hepp and Boccardo. To describe the transformation of austenite into ferrite in the bainite range, Hepp used the solution of Fick's second law using the finite difference method. In 2015, Boccardo and co-authors published a paper (Boccardo et al., 2015) in which they presented a thermo-mechanical-metallurgical model enabling the simulation of ausferritization of ductile iron. The model they proposed made it possible to predict the volume fraction of individual phases (graphite, ferrite, residual and residual austenite, martensite), the shape of changes, as well as residual stresses based on the initial microstructure, chemical composition and transformation temperature. The metallurgical model was based on Avrami's theory and allowed one to predict the volume fraction of the main phases occurring in the microstructure after the ausferritization process. As part of the presented research, an attempt was made to perform Particle Swarm Optimization (PSO) and with an Evolutionary Optimization (EO) algorithm to determine the parameters of a mathematical model describing the phase changes occurring in ductile iron during isothermal heat treatment. The results and effectiveness of both methods are presented in this article.

2. State of the art

ADI is a material with unique properties, combining high tensile strength and abrasion resistance with very good ductility. Due to these properties and relatively low production costs, it is used in many industries as a substitute for alloy cast steel and carburized or heat-treated steel. ADI owes its unique properties to the characteristic microstructure called ausferrite, which is shaped in the process of isothermal heat treatment in the bainite range. As research has shown (Nofal, 2013; Kuziak et al., 2010), the wide range of the mechanical properties of ADI can be obtained by appropriately

selecting isothermal strength parameters leading to the formation of the final microstructure. As part of the research, to calculate the bainite transformation time in ductile cast iron, the mathematical Bhadeshia model (Equation (1)) was used, which was originally used for high-silicon steels (Chester & Bhadeshia, 1997; Olejarczyk-Woźeńska et al., 2017):

$$t = \frac{\exp(C)}{A'(B+1)} \left\{ \exp(E)(\ln |1 + B\xi| + f(-(E + D\xi) - f(-E))) \right. \\ \left. - \exp(-D)(\ln(1 - \xi) + f(D(1 - \xi)) - f(D)) \right\} + \\ \frac{\exp\left(\frac{Q'}{R(T_{pi} + 273)} + C_4\right)(T_{pi} + 273)^z}{|\Delta G_m^0|^p} \quad (1)$$

where:

$$A' = \frac{1}{\theta \cdot L \cdot K_1}$$

$$B = \beta \cdot \theta$$

$$C = \frac{K_2}{R \cdot (T_{pi} + 273)} \left(1 + \frac{\Delta G_m^0}{r} \right)$$

$$D = \frac{K_2 \cdot (\Delta G_m^0 - G_N)}{r \cdot R \cdot (T_{pi} + 273)}$$

$$E = \frac{D}{B}$$

and:

λ_1, λ_2 – experimental constants; K_1', K_2 – experimental constants [J/mol]; $\xi = 0.01$ – relative volume fraction of bainite; $\theta = (1 - 0.1) \cdot (C_{gsr} - \bar{x}) / (C_{gsr} - 0.37)$ – the maximum volume fraction of bainite; $\beta = \lambda_1 \cdot (1 - \lambda_2 \cdot \bar{x}_m)$ – autocatalysis factor; $Q' = 243200$, $C_4 = -135$, $p = 5$, $z = 20$ – experimental constants; $K_1 = K_1' / u$ – austenite grain size function [1/(mm³·s)]; $u_t = 0.011$ – the length of the ferrite needles [μm³]; $u_w = 0.001077 \cdot T_{pi} - 0.2681$ – the width of the ferrite needles [μm³]; $u = u_t^2 \cdot (u_w \cdot 0.001)$ – unit volume of bainite ferrite [μm³]; $r = 2540$ – experimental constant [J/mol]; $R = 8.314$ – universal gas constant [J/(mol·K)]; $T_{pi} \in (280 - 400)$ – temperature of isothermal transformation [°C]; $T_{aus} = 420$ – austenitizing temperature [°C], $\Delta G_m^0 \in (-1377.71 - 2269.05)$ – initial value of the maximum possible change in the free energy of nucleation [J/mol]; $G_n = 3.637 \cdot T_{pi} - r$ – universal nucleation function [J/mol]; $\bar{x} = T_{aus} / 420 - 0.17 \cdot z_{Si} - 0.95$ – average carbon content in cast iron [mol%]; \bar{x}_m – average carbon content in carbon content in cast iron [%]:

$$\bar{x}_m = \frac{\left(\frac{\bar{x}}{W_c \cdot 100} \right)}{\left(\frac{\bar{x}}{W_c \cdot 100} + \frac{100 - \bar{x}}{WF_E \cdot 100} \right)};$$

$C_{gsr} = -0.0000033708 \cdot T_{pi}^2 + 0.0005114718 \cdot T_{pi} + 2.1581768606$ – average carbon content in austenite [%]; $W_c = 12$ – carbon atomic mass [u]; $WF_e = 55.85$ – iron atomic mass [u]; $\bar{L} = 80$ – average grain size of austenite [μm]; $z_{Si} = 2.55$ – silicon content in cast iron [mol]; n – number of iterations.

Some of the parameters included in the mathematical Bhadeshia model, i.e.: K_1' , K_2 , λ_1 and λ_2 are constants, the values of which are determined on the basis of time-consuming and expensive experiments. As part of the presented work, these parameters were identified using the PSO and EO algorithms, and the effectiveness of these methods was analyzed. The mathematical model and optimization algorithms were implemented in the C# programming language in the Visual Studio 2019 environment.

3. Research methodology

3.1. Identification of model parameters

The inverse calculation method was used to identify selected parameters of the bainite transformation time model in ductile cast iron. This method identifies the parameters of the model by searching for the minimum of the objective function based on the value of the model determined on the basis of previously conducted experimental studies (Mrzyglód et al., 2017). The objective function described by Equation (2) is the normalized mean square error (NMSE) between the bainite transformation time in the ductile cast iron, calculated on the basis of the mathematical Bhadeshia model (Equation (1)) and the time derived from dilatometric experiments (Tab. 1).

$$NMSE = \frac{\sum_{i=1}^n (t_{\text{exp } i} - t_{\text{comp } i})^2}{\sum_{i=1}^n (t_{\text{exp } i})^2} \quad (2)$$

where:

$t_{\text{exp } i}$ – i -th time of bainite transformation in ductile cast iron coming from the experiment (dilatometric tests) [s];

$t_{\text{comp } i} = f_i(K_1', K_2, \lambda_1, \lambda_2)$ – i -th time of bainite transformation in ductile cast iron calculated on the basis of the mathematical Bhadeshia model (Equation (1)) [s].

The ranges of the calculated coefficients were determined based on the articles (e.g. Chester & Bhadeshia, 1997; Rees & Bhadeshia, 1992), and their boundaries are presented in Table 1.

Table 1. The boundaries of the search area for each of the optimized parameters

Coefficient	x_{\min}	x_{\max}
K_1'	0.000000339	0.388
K_2	1.93	37700
λ_1	4.756	259.20
λ_2	0.00	39.69

During the experiment and during the calculations, the bainite transformation time was measured for different ausferritization temperatures (T_{pi}) and for different values of the maximum free energy of nucleation (ΔG_m^0) (Tab. 2).

Table 2. Time values of 99% bainite transformation in ductile iron and the maximum free energy of nucleation for different ausferritization temperatures

T_i [°C]	$t_{\text{exp } i}$ [s]	ΔG_m^0
400	5754.399	–1377.71
380	6760.83	–1520.32
360	8317.638	–1665.57
340	10010.61	–1813.27
320	10964.78	–1963.20
300	12882.5	–2115.19
280	13818.48	–2269.05
260	14971.28	–2424.60
240	16042.02	–2581.65

3.2. Particle Swarm Optimization (PSO) algorithm

Particle Swarm Optimization begins with the process of initiating the velocity dimensions and the position of all particles in the swarm with the help of random values coming from a specific search area. Once the initiation process is complete, each particle calculates its target function. If the target function of a given particle is better than the others, it and the position on the basis of which it was calculated are remembered globally. After all of the particles have computed the objective function, the main optimization stage begins, consisting of an iterative search of the solution space. In the course of this search, in each iteration, the particles determine their new velocities from Equation (3) (Dai et al., 2018):

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_{1i}(p_i - x_i(t)) + c_2 \cdot r_{2i}(p_g - x_i(t)) \quad (3)$$

where:

w – weight of inertia;

$v_i(t)$ – the current speed of the i -th particle;

r_{1i} , $r + 2i$ – random numbers from the interval [0, 1] computed separately for each component;

c_1, c_2 – cognitive and social learning coefficients;
 p_i – the best position of the i -th particle;
 p_g – best position of all particles.

By identifying new speeds, they then determine their new position based on the formula:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

where:

$x_i(t)$ – previous i -particle position;
 $v_i(t+1)$ – new position of the i -particle.

Due to the fact that particles are abstract entities, unlike their real counterparts they can occupy the same positions, so the issues related to avoiding collisions are ignored. If the new position of the particle is better than the initiated one, but also, in the course of the iteration procedure, it is better than any of its previously determined positions, then this position is remembered as the best position found so far by this particle. When all particles remember their new positions, they recalculate their target functions. The procedure for determining these functions is the same as in the initial step. After all iterations have passed or the stop condition is met, the PSO ends. The result of this optimization is the particle that achieves the best value of the objective function (Dai et al., 2018).

Then, the values of coefficients of inertia weight, cognitive and social learning were selected based on the literature (Wang et al., 2007). The coefficients were selected to ensure the convergence of the algorithm. In the case of the inertia weight, it could be a constant value (lower algorithm efficiency) or with a linear decrease (higher algorithm efficiency) – Equation (5) (Tab. 3) (Eberhart & Shi, 2001; Wang et al., 2007).

$$w_{LinearDecrease} = (w_{MAX} - w_{MIN}) \cdot \frac{(i_{max} - i_{current})}{i_{max}} + w_{MIN} \quad (5)$$

where:

w_{MAX} – the maximum value of the inertia weight (0.9);
 w_{MIN} – the minimum value of the inertia weight (0.4);
 i_{max} – total iteration number;
 $i_{current}$ – current iteration number.

In the case of cognitive and social learning, these could be constants with the same values (Tab. 3).

Table 3. Values of the weight of inertia and cognitive and social learning

Coefficient	Value
w_{const}	0.729
$w_{LinearDecreasing}$	(0.9–0.4)
c_1	1.49445
c_2	1.49445

Finally, the number of iterations and swarm particles were determined using random values ranging from 1 to 1000. After the identification of the initial settings of the PSO algorithm was completed, its implementation began.

3.3. Evolutionary Optimization (EO) algorithm

The Evolutionary Optimization algorithm is based on an iterative process of improving the solution sought, where the improvement is achieved by randomly searching the solution space, reproducing the best and mutating solutions. The result of this optimization is the individual whose objective function achieved the best value (McCaffrey, 2012). The method of selection and overcrossing based on randomization was used in this work. This method is responsible for the iterative search of the solution space, during which, in each iteration, a pair of parents is randomly determined from the entire population (Sumathi, 2008). This pair of parents is made up of two children (with a gene pool defined by a random crossing point), who are mutated and for whom the objective functions are calculated. Reproduction ends with the creation of new descendants. New descendants are then subjected to a mutation that changes the values of their genes, but not all, only a certain number of them, the size of which depends on the value of the mutation rate. The value of the mutation rate $mutationRate = 1/n$, where n is the number of optimized parameters, in the analyzed case $n = 4$. After the mutation is complete, the descendants calculate their target functions. When the objective functions are calculated, an additional individual is created known as an immigrant, whose task is to introduce a new pool of genes into the population (preventing the algorithm from getting stuck in local minima). This immigrant, along with the newly created descendants, is added to the population, replacing the same number of individuals in this population whose target functions were the worst. After adding new individuals to the population, it is checked whether the value of the objective function of one of them is better than the best one so far. If so, then the target function of such an individual becomes the new best. As soon as all iterations are completed, or the stop condition is present, the EO optimization is complete. The result of this optimization is the individual whose objective function achieved the best value.

4. Calculation results

The best result of the normalized mean square error (objective function) obtained by the PSO algorithm is 0.0057, for parameters $K_1' = 0.0000050$, $K_2 = 8436.89$, $\lambda_1 = 202.20$, $\lambda_2 = 19.213$.

The best result (NMSE – objective function) obtained using the EO algorithm was the error of 0.0057, for parameters $K_1' = 0.0000097$, $K_2 = 8010.362$, $\lambda_1 = 216.309$, $\lambda_2 = 11.7789$.

Figure 1 shows the results of the calculation of the transformation time using the optimized parameters and the results of the experiment. The diagram on the left is for the PSO algorithm and the diagram on the right is for the EO algorithm.

Based on the obtained results, the maximum NMSE was determined for the PSO and EO algorithms. It was assumed that the permissible maximum value of the objective function should be less than 0.006.

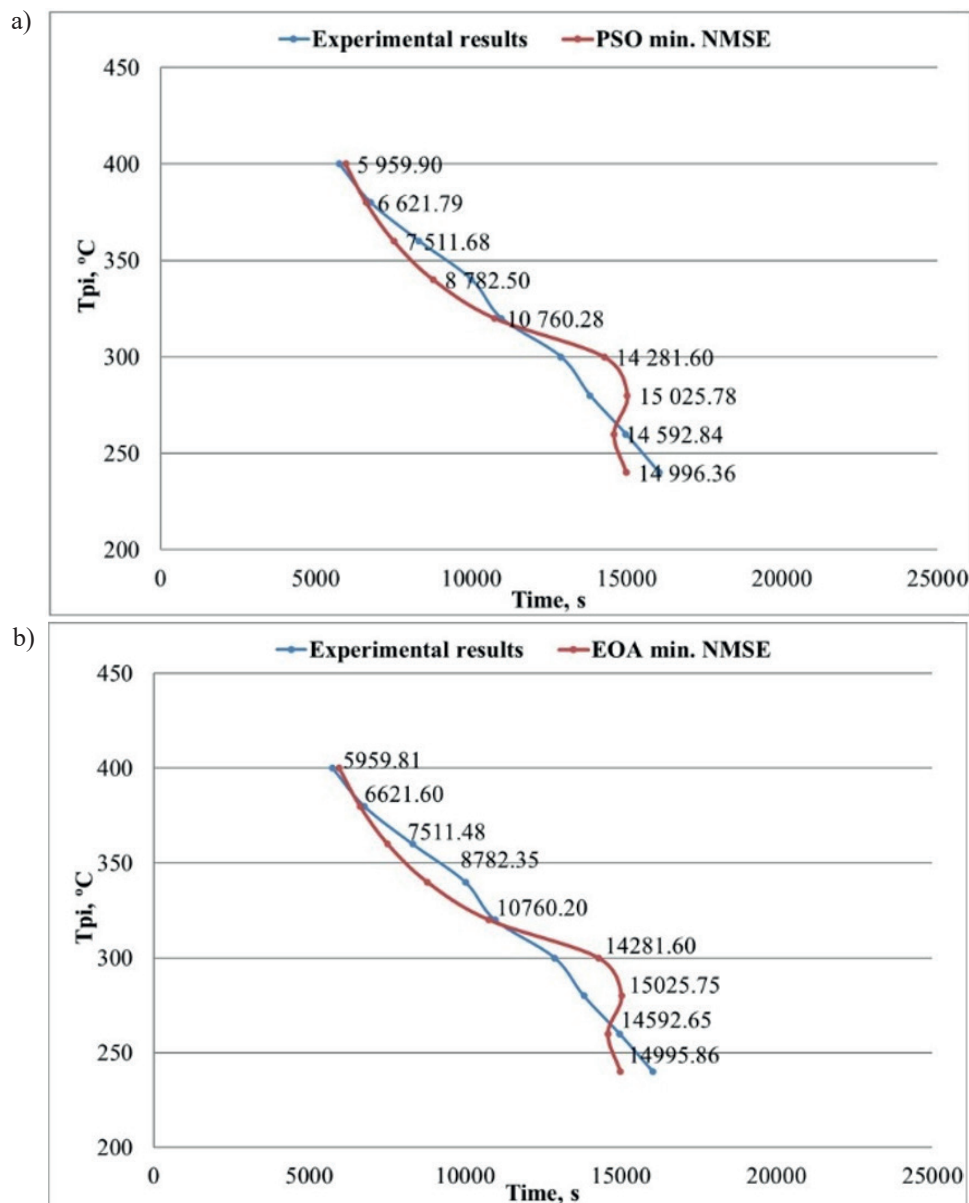


Fig. 1. Comparison of bainite transformation time in ductile cast iron at different ausferritization temperatures obtained with the PSO (a) and EO (b) algorithms and the experiment

Tables 4 and 5 show ten sets of identified parameters derived from the best optimization results

for the PSO algorithm (Tab. 4) and EO algorithm (Tab. 5).

Table 4. The values of the top ten parameters of the model optimized with the PSO algorithm

No.	K_1'	K_2	λ_1	λ_2	Goal function
1	5.01E-06	8436.892	202.2037	19.21305	0.00572809785
2	4.98E-06	8440.847	142.0599	15.80991	0.00572809790
3	4.94E-06	8446.756	147.7473	16.35244	0.00572809843
4	5.07E-06	8427.809	93.19192	9.377675	0.00572809870
5	4.92E-06	8448.681	106.2462	12.08332	0.00572809875
6	5.08E-06	8426.808	134.9443	14.93806	0.00572809887
7	5.08E-06	8425.532	85.37225	7.686909	0.00572809905
8	5.09E-06	8424.932	154.6197	16.49391	0.00572809925
9	5.10E-06	8424.271	146.2274	38.96477	0.00572809939
10	5.10E-06	8423.901	82.78194	6.990056	0.00572809947

Table 5. The values of the top ten parameters of the model optimized with the EO algorithm

No.	K_1'	K_2	λ_1	λ_2	Goal function
1	9.73E-06	8010.362	216.309	11.77898	0.00572962869
2	1.27E-05	7895.355	178.947	2.178666	0.00573114332
3	1.83E-05	7645.972	252.2327	1.200476	0.00573305373
4	3.26E-06	8576.834	88.09056	39.06593	0.00573335937
5	5.54E-06	8169.649	248.8016	34.73644	0.00573339272
6	1.67E-05	7794.549	221.4534	0.133712	0.00573347868
7	1.30E-05	8009.958	236.079	7.7524	0.00573511829
8	1.79E-05	7526.145	235.5786	0.247519	0.00573650762
9	1.69E-05	7712.015	257.2354	3.734918	0.00573713206
10	1.15E-05	8137.582	212.2943	8.070432	0.00573802421

Based on the values from Table 5, it was found that the global minimum obtained using the PSO algorithm, with a constant number of particles equal to 10 and a constant number of iterations equal to 1000, is achieved when the values of the optimized parameters are within the following ranges: $K_1' \in (0.00000492 - 0.00000510)$, $K_2 \in (8423.901 - 8448.681)$, $\lambda_1 \in (82.78194 - 202.2037)$ and $\lambda_2 \in (6.990056 - 38.96477)$.

Based on the data contained in Table 6, it was found that the global minimum obtained by the EO algorithm, with a constant number of individuals equal to 10 and a constant number of iterations equal to 1000, is achieved when the values of the optimized parameters are within the following ranges: $K_1' \in (0.00000326 - 0.0000183)$, $K_2 \in (7526.145 - 8576.834)$, $\lambda_1 \in (88.09056 - 257.2354)$ and $\lambda_2 \in (0.133712 - 39.06593)$.

5. Testing the effectiveness of models

In order to test the effectiveness of the analyzed optimization models, a number of tests were carried out. The influence of the number of particles and the number of iterations on the algorithm convergence was analyzed.

The number of iterations necessary to perform was tested and the number of calls to the objective function for both algorithms was compared. Based on the conducted tests, the best optimization strategies for the problem being solved were selected.

5.1. Influence of the number of iterations on the convergence of identification

Based on the performed calculations, it was found that the PSO algorithm achieved 98.7% coverage, while the EO algorithm achieved 6.8% coverage. Then, tests were carried out to check the influence of particles (in PSO algorithm) and individuals (in EO algorithm) and the number of iterations on the normalized mean square error. Figure 2 shows the results of the analysis with the PSO and EO algorithms, respectively for 5 particles or individuals and a variable number of iterations: 10, 100 and 1000.

Figure 3 shows the results of the analysis with the PSO and EO algorithms, respectively, for 10 particles and individuals and a variable number of iterations: 10, 100 and 1000.

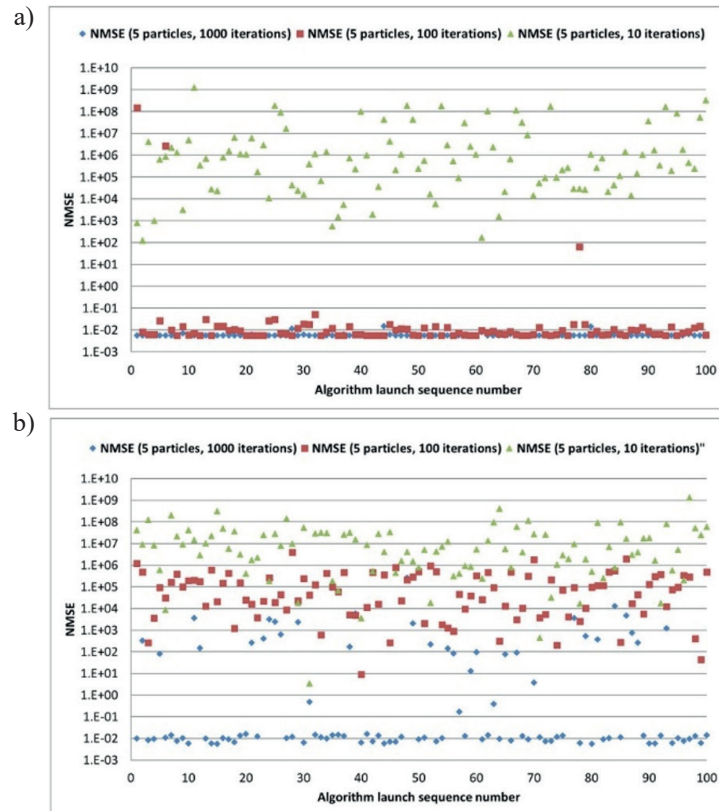


Fig. 2. Effect of the number of iterations on the convergence of PSO (a) and EO (b) algorithms with a constant number of particles equal to 5

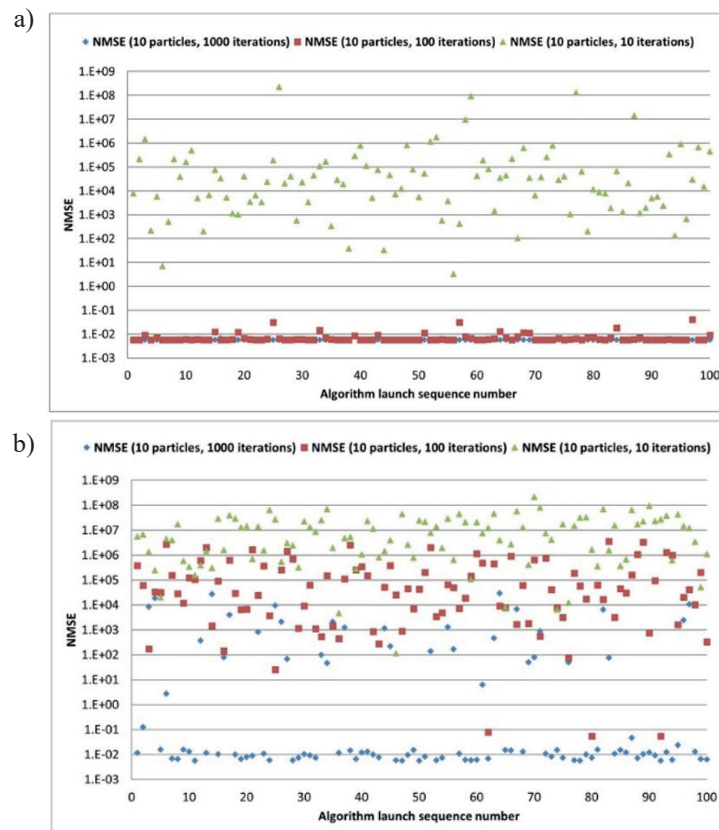


Fig. 3. Effect of the number of iterations on the convergence of PSO (a) and EO (b) algorithms with a constant number of particles equal to 10

Based on the data illustrated in Figure 2 and Figure 3, it can be concluded that the number of iterations affects the convergence of both algorithms. The normalized mean square error obtained with the PSO and EO algorithms at a constant number of particles decreases with an increasing number of iterations. The magnitude of this decrease depends on the type of algorithm and is greater for the PSO. In the case of the PSO algorithm, the results were convergent for 100 and 1000 iterations. However, in the case of EO algorithm, this was only true for 1000 iterations.

5.2. The influence of the number of particles and individuals on the coincidence of identification

Another test concerned the influence of the number of particles and individuals (with a constant number of iterations) on the calculation results. The analyzes were carried out for a variable number of particles: 3, 5, and 10, and for two iteration variants: 100 and 1000. Figure 4 shows the results of convergence for 3, 5 and 10 particles and for 100 iterations for the PSO and EO algorithms.

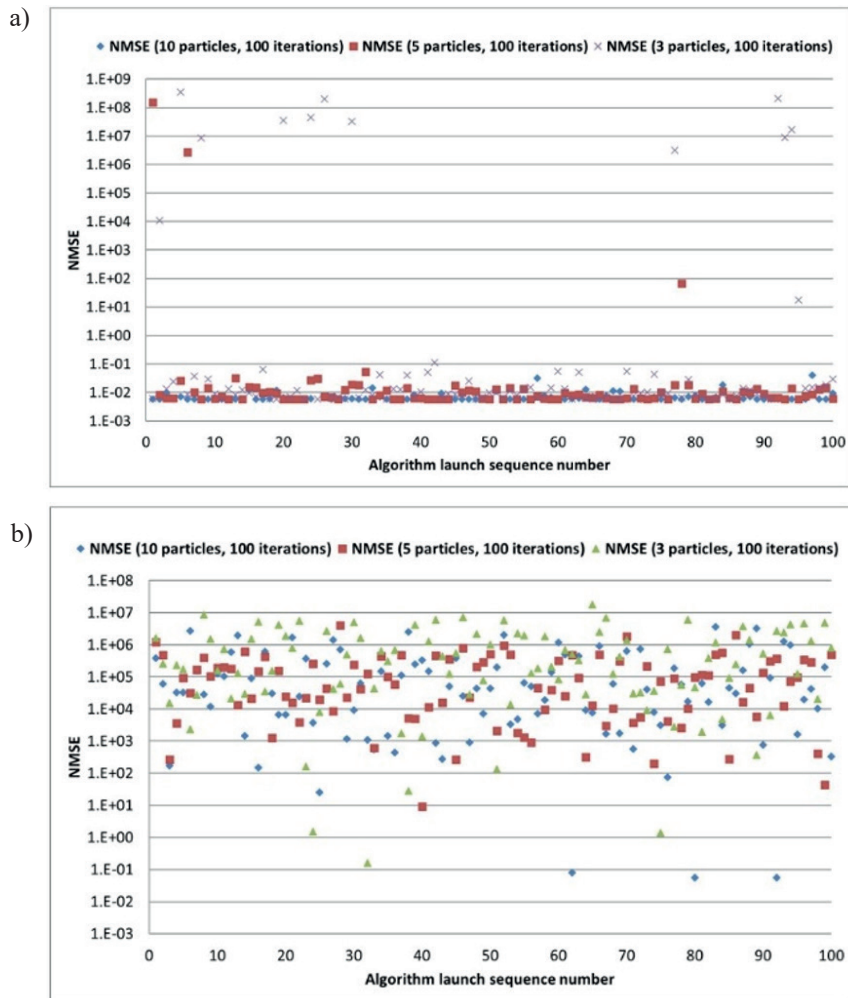


Fig. 4. Influence of the number of individuals on the convergence of PSO (a) and EO algorithms (b) with a constant number of iterations equal to 100

The results shown in Figure 4 showed that for the PSO algorithm, convergence was achieved for both 10, 5 and 3 particles. However, in the case of the EO algorithm, convergence was not achieved even at 100 iterations.

Figure 5 shows the convergence of calculations for 1000 iterations and 3, 5 and 10 particles for the PSO and EO algorithms.

The results shown in Figure 5 confirmed that the normalized mean square error obtained by the PSO and EO algorithms, with a constant number of iterations equal to 1000, decreases with an increasing number of particles and individuals. Much better results were obtained for the PSO algorithm, where very good convergence was obtained for 3, 5 and 10 particles. In the case of EO for 1000 iterations, the convergence was noticeable for 5 and 10 subjects.

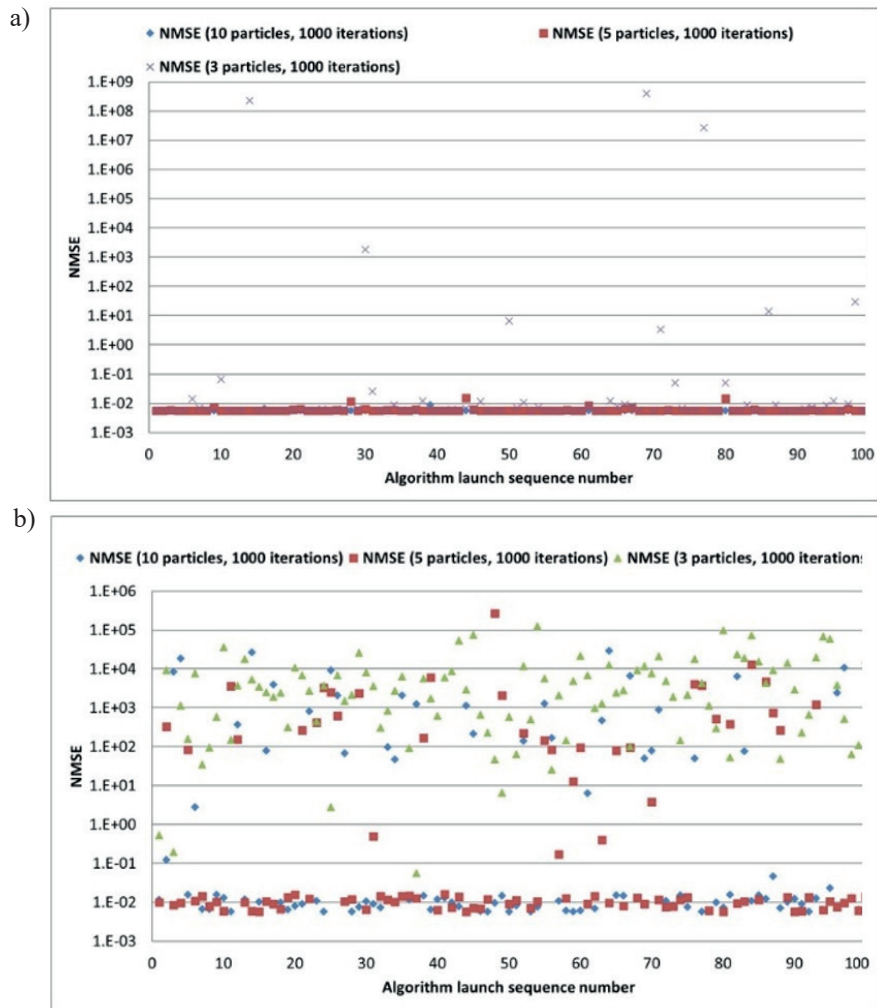


Fig. 5. Influence of the number of individuals on the convergence of PSO (a) and EO (b) algorithms with a constant number of iterations equal to 1000

5.3. Testing the necessary number of iterations of algorithms and the number of calls to the objective function

Subsequent tests were performed to determine the minimum number of iterations required to obtain the global minimum for 5 and 10 particles and individuals. During this test, both algorithms were run 100 times and the convergence was analyzed for each call. Based on the test results, it can be noticed that the PSO algorithm needs an average of 411 iterations for 5 particles, and an average of 285 iterations for 10 particles to achieve the global minimum, while the EO algorithm for 5 individuals needs approximately 700, and for 10 individuals, approximately 600 (although it does not always converge). The conducted analysis confirms that the number of iterations needed to determine the global minimum decreases as the number of particles increases.

Figure 6 presents a comparison of the number of calls to the objective function for both algorithms, for 5 and 10 particles and individuals.

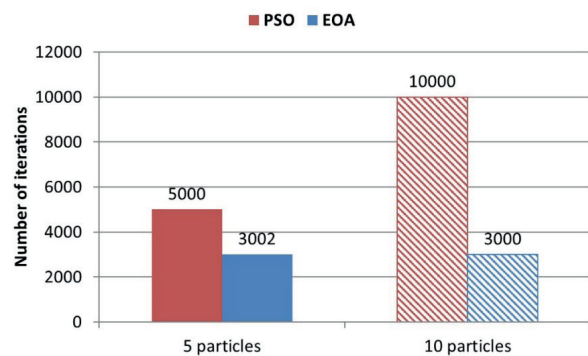


Fig. 6. The minimum number of iterations needed to determine the global minimum using the PSO and EO algorithms (for 100 runs) and with a constant number of particles and individuals equal to 5 and 10

The analyzes carried out in terms of the number of calls to the objective function showed that for both 5 and 10 particles and individuals, EO algorithm shows a much smaller number of calls to the objective function than PSO.

5.4. Determining the best optimization strategies

Table 6 shows the percentage convergence of the analyzed algorithms depending on the number of iterations and the number of particles and individuals. It was assumed that the algorithm obtains a satisfactory convergence if the value of the objective function is less than 0.006 (the adopted maximum allowable value of the objective function).

Table 6. Percentage convergence of the PSO and EO algorithms depending on the number of particles and the number of iterations

Iterations	PSO [%]		
	3 particles	5 particles	10 particles
10	0	0	0
100	16	31	62
1000	59	86	98
Iterations	EO [%]		
	3 individuals	5 individuals	10 individuals
10	0	0	0
100	0	0	0
1000	0	7	11

Based on the results presented in Table 6, it was determined that the best strategies could only be obtained for the PSO algorithm for 5 particles and 1000 iterations (86%) and for 10 particles and 1000 iterations (98%).

The EO algorithm did not converge above 50% for any of the configurations tested.

6. Summary

As part of the presented research, two optimization algorithms (PSO, EO) were developed and implemented, allowing for the satisfactory determination of the coefficients of the model of phase transformations occurring in ductile cast iron during isothermal heat treatment in the field of bainite transformation. The obtained results give the correct values in relation to the experimental values. The parameters of the developed model were identified on the basis of inverse analysis, experimental research and literature data. The inverse analysis was performed

based on the results of dilatometric tests. The PSO and EO methods were used to optimize the objective function.

The conducted research revealed that there are configurations of the number of particles and individuals and the number of iterations that ensure the determination of the identified parameters. The convergence of both methods largely depends on the number of particles and individuals as well as the number of iterations. On the basis of the conducted analyzes, it was found that the objective function minimized with the use of both algorithms, with a constant number of particles, decreases with an increase in the number of iterations. The magnitude of this decrease depends on the type of algorithm and is much greater for the PSO method.

Tests for a fixed number of iterations and a variable number of particles and individuals also showed the effect of this parameter on the algorithm's convergence. The tests carried out revealed that in the case of PSO, satisfactory results were achieved for 5 particles and 100 and 1000 iterations, and for 10 particles and 100 and 1000 iterations. However, in the case of EO, satisfactory convergences were obtained only for 5 individuals and 1000 iterations, and for 10 individuals and 1000 iterations.

For the analyzed problem, the PSO algorithm with the used parameters shows a much better convergence. However, it would definitely be worth testing other parameters of these models during subsequent tests. With 1000 iterations for 100 runs of the algorithm, more than 50% convergence was achieved for both 10, 5 and 3 particles. In addition, the PSO algorithm achieved 86% convergence for 1000 iterations and 5 particles, and for almost 100% convergence (98%) for 10 particles. On the other hand, EO did not converge to more than 50% for any of the analyzed configurations (number of particles – number of iterations).

The analysis of the number of calls to the objective function for 1000 iterations showed that the PSO algorithm for 5 and 10 particles triggers this function much more often than EO. The conducted research allowed to select the best optimization strategies for the presented problem. The best result obtained using the PSO algorithm was characterized by an NMSE error of 0.005728, for which the optimized coefficients were: $K_1' = 0.0000050$, $K_2 = 8436.89$, $\lambda_1 = 202.2037$, $\lambda_2 = 19.213$. However, for the EO algorithm, the best result was an NMSE error of 0.005729, for which $K_1' = 0.0000097$, $K_2 = 8010.36$, $\lambda_1 = 216.309$, $\lambda_2 = 11.77$.

Acknowledgements

The work has been supported by the Polish Ministry of Science and Higher Education – AGH University of Science and Technology Funds no. 16.16.110.663.

References

- Boccardo, A.D., Dardati, P.M., Celentano, D.J., Godoy, L.A., Górny, M., & Tyrała E. (2015). Numerical simulation of austempering heat treatment of ductile cast iron, *Metallurgical and Materials Transactions B*, 47(1), 566–575. <https://doi.org/10.1007/s11663-015-0511-y>.
- Chester, N.A., & Bhadeshia, H.K.D.H. (1997). Mathematical modelling of bainite transformation kinetics. *Journal de Physique*, 07(C5), C5-41–C5-46. <https://doi.org/10.1051/jp4:1997506>.
- Dai, H.-P., Chen, D.-D., & Zheng, Z.-S. (2018). Effects of random values for Particle Swarm Optimization algorithm. *Algorithms*, 11(2), 23. <https://doi.org/10.3390/a11020023>.
- Eberhart, R.C., & Shi, Y. (2001). Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)* (pp. 81–86, vol. 1). <https://doi.org/10.1109/CEC.2001.934374>.
- Essaid, M., Idoumghar, L., Lepagnot, J., Bréviliers, M., & Fodorean, D. (2018). *A hybrid optimization algorithm for electric motor design*. In Y. Shi, H. Fu, Y. Tian, V.V. Krzhizhanovskaya, M.H. Lees, J. Dongarra, P.M.A. Sloot (Eds.), *Computational Science – ICCS 2018. 18th International Conference, Wuxi, China, June 11–13, 2018, Proceedings* (part II), Springer Cham. https://doi.org/10.1007/978-3-319-93701-4_39.
- Gili, M., Maringer, D., & Schumann, E. (2019). *Numerical Methods and Optimization in Finance* (2nd ed.). Elsevier–Academic Press.
- Hepp, E., Hurevich, V., & Schäfer, W. (2012). Integrated modeling and hest treatment simulation of austempered ductile iron. *IOP Conference Series: Materials Science and Engineering*, 33, 1–10. <https://doi.org/10.1088/1757-899X/33/1/012076>.
- Kuziak, R., Zalecki, W., & Pietrzyk, M. (2010). Matematyczne modelowanie hartowności bainitycznej. *Prace Instytutu Metalurgii Żelaza*, 62(1), 27–32.
- McCaffrey, J. (2012). Test run – evolutionary optimization algorithms. *MSDN Magazine*, 27(6).
- Mrzygłód, B., Kowalski, A., Olejarczyk-Woźńska, I., Giętka, T., & Głowacki, M. (2017). Characteristics of ADI ductile cast iron with single addition of 1.56% Ni. *Archives of Metallurgy and Materials*, 62(4), 2273–2280. <https://doi.org/10.1515/amm-2017-0335>.
- Nofal, A. (2013). Advances in the metallurgy and applications of ADI. *Journal of Metallurgical Engineering*, 2(1), 1–18.
- Olejarczyk-Woźńska, I., Adrian, H., Mrzygłód, B., & Głowacki, M. (2017). Mathematical model of bainitic transformation in austempered ductile iron. *Archives of Foundry Engineering*, 17(4), 200–206. <https://doi.org/10.1515/afe-2017-0158>.
- Rees, G.I., & Bhadeshia H.K.D.H. (1992). Bainite transformation kinetics. Part 1. Modified model. *Materials Science and Technology*, 8(11), 985–993. <https://doi.org/10.1179/mst.1992.8.11.985>.
- Sumathi, S., Hamsapriya, T., & Surekha, P. (2008). *Evolutionary Intelligence. An Introduction to Theory and Applications with Matlab*. Springer-Verlag Berlin Heidelberg.
- Tang, L., Zhao, Y., & Liu, J. (2013). An improved differential evolution algorithm for practical dynamic scheduling in steelmaking-continuous casting production. *IEEE Transactions on Evolutionary Computation*, 18(2), 209–225. <https://doi.org/10.1109/TEVC.2013.2250977>.
- Wang, Z., Sun, X., & Zhang, D. (2007). *A PSO-based classification rule mining algorithm*. In Huang, D.S., Heutte, L., & Loog, M. (Eds.), *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues. Third International Conference of Intelligent Computing, ICIC 2007, Qingdao, China, August 21–24, 2007 Proceedings* (pp. 377–384). Springer Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74205-0_42.
- Zimba, J., Henwood, D., Navara, E., & Simbi, D.J. (1999). Three-dimensional diffusion model for austenitization of ferritic spheroidal graphite irons. *Materials Science and Technology*, 15(9), 1024–1030. <https://doi.org/10.1179/026708399101506887>.

