DOI 10.2478/ama-2022-0039

SELECTION OF THE HEAT TRANSFER COEFFICIENT USING SWARMING ALGORITHMS

Elżbieta GAWROŃSKA*, Robert DYJA*, Maria ZYCH*, Grzegorz DOMEK**

*Faculty of Mechanical Engineering and Computer Science, Czestochowa University of Technology,
ul. Dąbrowskiego 73, 42-201 Częstochowa, Poland
**Faculty of Mechatronics, Kazimierz Wielki University, ul. Kopernika 1, 85-074 Bydgoszcz, Poland

elzbieta.gawronska@icis.pcz.pl, robert.dyja@icis.pcz.pl, maria.zych@icis.pcz.pl, gdomek@ukw.edu.pl

received 24 June 2022, revised 9 August 2022, accepted 10 August 2022

Abstract: The article presents the use of swarming algorithms in selecting the heat transfer coefficient, taking into account the boundary condition of the IV types. Numerical calculations were made using the proprietary TalyFEM program and classic form of swarming algorithms. A function was also used for the calculations, which, during the calculation, determined the error of the approximate solution and was minimalised using a pair of individually employed algorithms, namely artificial bee colony (ABC) and ant colony optimisation (ACO). The tests were carried out to select the heat transfer coefficient from one range. Describing the geometry for a mesh of 408 fine elements with 214 nodes, the research carried out presents two squares (one on top of the other) separated by a heat transfer layer with a κ coefficient. A type III boundary condition was established on the right and left of both edges. The upper and lower edges were isolated, and a type IV boundary condition with imperfect contact was established between the squares. Calculations were made for ABC and ACO, respectively, for populations equal to 20, 40 and 60 individuals and 2, 6 and 12 iterations. In addition, in each case, 0%, 1%, 2% and 5% noise of the reference values were also considered. The obtained results are satisfactory and very close to the reference values of the κ parameter. The obtained results demonstrate the possibility of using artificial intelligence (AI) algorithms to reconstruct the IV type boundary condition value during heat conduction modelling.

Key words: swarm algorithm, ABC algorithm, ACO algorithm, heat transfer coefficient, computer simulation, numerical modelling

1. INTRODUCTION

Artificial intelligence (AI) is a branch of computer science of a practical and cognitive nature, and finds a steadily increasing volume of applications not only in science, technology and engineering but also in everyday life. This section covers re-search on intelligent systems, their modelling, construction and use to support and substitute human mental work and deepen understanding of human reasoning. AI methods are irreplacea-ble in a situation where it is necessary to infer further proce-dures based on incomplete information on a given issue. One of the elements of AI is optimisation issues. Optimisation problems appear in almost every area of science, engineering, economics and other fields of study. In order to solve most of today's optimisation problems, it is necessary to use algorithms that adapt easily to constraints and do not depend on the num-ber of variables and the size of the solution space. Then, vari-ous types of nature-/biology-inspired algorithms come in handy, such as genetic algorithm (GA), differential evolution (DE), ant colony optimisation (ACO) and artificial bee colony (ABC), the rules of which are taken from observation of nature. In recent years, there has been an increase in interest in the class of algorithms called swarms; these algorithms are based on swarm intelligence. Their use has made it possible to signifi-cantly improve the performance of a given activity through its far-reaching optimisation. Even though the first optimisation algorithms were developed in the previous century, research-ers still exhibit great interest in this topic, especially with regard to optimisation of issues related to heat transfer [1,2].

AI algorithms have found many uses in solving various problems. For example, in the problem of image contrast enhancement, the ABC algorithm allowed obtaining better-quality images [3]. The use of the ACO algorithm improved the Elman neural network, thus leading the way for the capability to determine the state of charge of lithium-ion batteries in electric vehicles. Nature-inspired algorithms are also used to navigate mobile robots that have to navigate over uncertain terrain [4].

Karaboga et al. presented a new approach to solving the inverse heat conduction problem and estimating an unknown heat source. The author formulated the problem of physical heat transfer as an optimisation problem. The ABC algorithm, based on the intelligent behaviour of a honey bee swarm, turned out to be very simple and flexible compared to the existing swarming algorithms. The author showed that the algorithm is very stable for testing problems. The proposed algorithm can be used for unimodal and multimodal solving of numerical optimisation problems. The features of the algorithm proposed by Karaboga became the basis for the present authors' work in this article [5,6,7].

Physical problems modelled by mathematical models can be divided into direct and inverse problems [8]. Direct problems refer to a situation when all input data are known, and the problem is solved (mainly numerically) with initial-boundary assumptions [9]. Inverse problems consist of recreating some model parameters based on the experiment, which makes it possible to control its course and the final quality of the product [10,11]. This type of control is becoming more and more desirable in manufacturing processes. Solving the inverse heat conduction problem is more complex than solving the direct problem. Except in the most uncomplicated cases, it is impossible to solve the inverse problem with the help of analytical methods. Moreover, even if such a

DOI 10.2478/ama-2022-0039

solution exists, it is neither unequivocal nor stable. Therefore, the approximate method that gives a satisfactory solution is needed [12,13]. Conducting experimental research is laborious, time-consuming and requires additional financial outlays. Such an experiment can be particularly burdensome, for example, when a qualitative analysis of different materials needs to be performed. Using efficient and optimal numerical methods makes it possible to perform an extensive series of tests with relatively lower amounts of effort and cost involved.

Numerical simulation of heat transfer processes is based on solutions to both direct and inverse problems. The second approach has been very popular among researchers in recent years. An example of the use of inverse problems and optimisation algorithms to reconstruct the conditions of the experiment is the study by Matsevityi et al. [14], in which the authors successfully reconstructed the heat transfer coefficient with the environment (Newton boundary condition) and obtained graphs of temperature changes over time that were very consistent with those obtained from experimentation.

In the example presented in this article, selected optimisation swarm algorithms (ABC and ACO) are used to reconstruct the heat transfer coefficient between the cast and the casting mould in numerical modelling of the heat conduction problem.

2. METHODS

2.1. Heat conduction problem

The heat transfer process is divided into: steady-state, when the temperature distribution in the system under consid-eration does not change with time, and the amount of heat is constant (in this case, the only variables are coordinates in space); and transient (unsteady-state), when the temperature distribution and heat change with respect to time. The transient heat flow is considered in this paper.

During the heating and cooling process, unsteady-state heat conduction occurs when both bodies strive to achieve a temperature equilibrium with the environment in which they find themselves. According to Fourier's law, the heat flux conduc-tion density is directly proportional to the temperature gradient. The mathematical formula of heat transfer is defined as follows:

$$c\rho\frac{\partial T}{\partial t}+\nabla\cdot(-\lambda\nabla T)=Q \tag{1}$$

where *c* is specific heat $\left[\frac{J}{kgK}\right]$, ρ is density $\left[\frac{kg}{m^3}\right]$, *T* is temperature [K], *t* is time [*s*], λ is heat transfer coefficient $\left[\frac{W}{mK}\right]$ and Q represents internal heat sources $\left[\frac{W}{m^3}\right]$. Since the phase transformation was not taken into account, Q was equal to 0.

The problem of transient heat conduction is one of the initial-boundary issues, requiring the setting of appropriate initial and boundary conditions at the commencement of calculations. Initial conditions, called Cauchy conditions, allow giving specified temperature values at the initial instant for t = 0 s.

$$T(\mathrm{r},t)|_{t=0}=T_0(\mathrm{r}) \tag{2}$$

where r is the field vector at a given point. There are four types of boundary conditions that are associated with heat transfer:

- Dirichlet boundary condition (the first type condition) assumes the temperature (T_z) on the Γ boundary of area Ω.

  $$\Gamma: T=T_z. \tag{3}$$

  when the measurement's body surface temperature T_z is known, the boundary conditions can be formulated as Dirichlet boundary conditions.
- Von Neumann boundary condition (the second type condition) assumes the known heat flux (q_z) on the Γ boundary of area Ω.

  $$\Gamma: q=q_z. \tag{4}$$
- Newton boundary condition (the third type condition) assumes the exchanged heat with the environment on the boundary Γ of the area Ω.

  $$\Gamma: q=\propto(T-T_{amb}), \tag{5}$$

  where $\propto$ is the coefficient of heat exchange with the environment, *T* is the temperature of the body on the boundary Γ, T_{amb} is the ambient temperature and *q* is the heat flux flowing into the Ω area when $T<T_{amb}$ or flowing from the Ω area when $T>T_{amb}$. In the third boundary condition, the heat transfer coefficient determines the intensity of convection. The $\propto$ coefficient depends on the heat exchange type as well as the speed and direction of the flow to the body surface.
- Continuity condition (the fourth type condition) assumes contact and the heat exchange between two bodies. This contact may be ideal or non-ideal. In the non-ideal case, there is a separation layer with a κ coefficient:

  $$\kappa=\frac{\lambda_p}{\delta}, \tag{6}$$

  where λ_p is the thermal conductivity coefficient of the separation layer and δ is the thickness of this layer.

Using the previously described boundary conditions in the tested model, we find that the results are consistent with Fig. 3, and the area dimensions are given in Fig. 2. All boundaries (numbered from 1 to 8) are 0.02 m long.

This work aims to reconstruct the heat transfer coefficient κ through the layer separating the cast and the casting mould.

2.2. ABC algorithm

The first works on simulating swarming behaviour, using mathematical and numerical modelling, appeared in the 70s and 80s of the previous century. However, they started to be widely used only in the first decade of the 21st century. For example, the ABC algorithm uses the intelligent behaviour of a swarm of honeybees. The algorithm model is based on the search for food by an ABC.

The algorithm ABC consists of three main components: food sources, unemployed bees and employed bees closely related to the food source. The number of solutions in the population equals the number of food sources. The amount of nectar in the food source is the value of the evaluating function associated with the solution. In searching for food, bees have developed various techniques to improve communication with other bees about the location of food sources. Employed bees (scout bees) randomly choose the direction and area to search for the best available food sources. Once such a source is found, the scout becomes an employed bee and returns to the hive. They inform the bees that are spectators about the direction, distance and area of the food.

The information is passed on in the dance. After that, the sources are exploited, and the employed bees become unemployed when exhausted [15].

In the ABC algorithm, the position of the food source is a possible solution to the optimisation problem, and the nectar content of the food source corresponds to the quality (efficiency) of the related solution. The number of employed and unemployed bees equals the number of solutions in the population. In the first step, ABC generates a random initial population *P* of *SN* solutions, where *SN* is the number of the food source. Each solution x_i (i = 1, 2, ..., *SN*) is a vector of solutions to *D*. In the algorithm, *D* denotes the number of optimisation parameters. The determination of the food sources' coordinates is subjected to multiple iterations *C* = 1, 2, …, C_{max}, where the iterations mean the update of the solutions. The employed bee updates the changes in position (solution) depending on local information (visual information) and tests the amount of nectar (efficiency) of the new source (new solution). If the amount of new nectar is higher than in the previous iteration, the bee remembers the new nectar position and forgets the current one. Otherwise, it retains the previous position in its memory. After the search process is complete, all employed bees share nectar information from the various food sources and their position information with the bees in the dance area. An unemployed artificial bee selects a food source depending on the probability value p_i calculated according to the following formula:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \qquad \text{for } i = 1, 2, \dots, P, \tag{7}$$

where fit_i is an efficiency of i^{th} solution, proportional to the nectar amount of food source, and *SN* equals the number of the employed bees.

In the next step, the update of the coordinates of the food sources is carried out, which is based on the own coordinates of the bee and other employed bees. The above process follows the relationship:

$$v_i = x_{ij} + rand[-1, 1] \cdot (x_{ij} - x_{kj}) \tag{8}$$

where v_{ij} is the update of the food sources' coordinates, and *i, k* = 1, 2, …, *SN* and *j* = 1, 2, …, *D* are randomly selected within the given range. The variable *k* is determined randomly, too, and it must be different from *i*. The search for the optimal solution is related to the iterative process of reducing the difference between successive updates of the position of food sources.

The ABC algorithm uses four control parameters: (i) the global selection process used by unemployed bees Eq. (7), (ii) a local selection process carried out by the viewer bees and depends on local information about the adjacent food source Eq. (8), (iii) a local selection process, called sparse, carried out by all bees, and (iv) a random selection process carried out by the scout bees.

The survival and progress of a bee colony depend on the quick discovery and efficient use of the best food resources. Similarly, successful solutions to complex engineering problems are related to finding reasonable solutions relatively quickly, especially for problems that need to be solved in real-time.

2.3. ACO algorithm

The ACO algorithm was presented and developed by Marco Dorigo et al. in the 90s of the previous century and is a part of metaheuristic algorithms. The inspiration comes from the world of ants that can find the shortest route between an anthill and an available food source.

In the beginning, ants walking towards a food source choose the route randomly. They return to the anthill and leave a pheromone trail on their route, which gradually evaporates if other ants do not follow the path. On a shorter route, evaporation is slower than on longer routes, and thus subsequent ants choose this route more willingly than other routes, and by choosing it, they strengthen the pheromone trail. This phenomenon is called positive feedback. In an ACO algorithm, a colony of artificial individuals cooperates to search for optimal solutions to complex combinatorial problems. There is an indirect interaction between ants collecting some kind of experience and using it in further research. Each of the ants follows an identical strategy of finding the shortest path to the goal. Over time, the ants work together to work out a set of shortest paths leading them to their designated food sources. This is a manifestation of collective intelligence [16,17].

Finding the best solution in the classical ACO algorithm is an iterative process, too, similar to finding the best solution in the ABC algorithm. At each iteration, an ant selects a path to a food source depending on the probability value p_{ij} calculated according to the following formula:

$$p_{ij} = \frac{a_{ij}}{\sum_{l \in N_i} a_{il}}, \tag{9}$$

where N_i is the feasible neighbourhood of i^{th} node for k^{th} ant and is the set of nodes that she has not yet visited. At each iteration, a k^{th} ant located in the i^{th} node on the path to a food source chooses the j^{th} node from the nearest neighbourhood [18].

In order to avoid an unlimited increase in the pheromone traces, pheromone pairing is added in each t^{th} iteration in the algorithm. An additional matrix is created for each ant to store the nodes of the current traces. This matrix is called the pheromone array. In the first iteration, in the classic form of an algorithm, for each ant, the nodes on the paths from anthill to food sources are fixed by roulette wheel law after considering the probability determined by the formula in Eq. (9). It allows for the initiation of each ant's trace transition matrices. After going over all the ants, the trace with the best quality index is found for each iteration. Based on this indicator, the trace for each ant is modified. Then, for the best transition path (the superscript of best), new nodes are randomly determined as a particular random deviation from the position of the previous nodes:

$$x_{ij}(\mathrm{t}) = x_{ij}(t-1) + rand[0, 1] \cdot \left[x_{ij}^{best}(t-1) - x_{ij}(t-1)\right] \tag{10}$$

The above-mentioned algorithm steps are repeated in each iteration. Consequently, the traces determined in successive iterations get closer to the path characterised by the best quality indicator that all ants follow.

3. IMPLEMENTS AND ASSUMPTIONS

This article deals with a topic that requires the connection of two separate fields: thermomechanics and computer science. The cooling of the aluminium alloy has been selected from the thermomechanics field. The reconstruction of the value of one of the coefficients in the contact boundary condition became the main point of this work. The heat transfer coefficient of the layer separating the cast and the casting mould is selected based on the available literature, as well as according to the discretion of the person performing numerical calculations or the employee responsible for making the cast in the foundry, who makes an ideal decision based on their field experience and the prevalent circumstances. In this study, an attempt was made to fill the gap in this research area by creating a tool enabling the reconstruction of the heat transfer coefficient.

Finite element models and meshes were created in GMSH open source 3D finite element mesh generator [19]. Numerical calculations were made using an in-house software, whose application was combined with that of TalyFEM library, an application developed previously in the literature [20,21,22], as well as using algorithms implemented in the C++ language. The finite element method facilitates the modelling of complex problems, and its wide application allows for easy adaptation of many existing computational techniques. TalyFEM is a library that uses the finite element method to simulate selected physical phenomena, containing many PETSc (the Portable, Extensible Toolkit for Scientific Computation) data structures such as vectors, matrices and ready-made solvers [23]. The tests were performed on a computer with the following parameters – Processor: Intel (R) Core^{TM} i5-4590 CPU @ 3.30 GHz, x86_64 architecture; Manufacturer ID: Genuine Intel; CPU family: 6; CPU clock speed: 3279.890 MHz; Operating system: Linux (in distribution Ubuntu). In the field of AI, optimisation heuristic algorithms were used. These algorithms have been implemented in Python and adapted to the possibility of connecting them with TalyFEM [24].

The functional Eq. (11), determining the error of the approximate solution, and minimised with the use of selected swarming algorithms, assumes the form of the formula:

$$J(\kappa) = \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\left(T_{ij} - U_{ij}\right)^2, \qquad (11)$$

where N_1 is the number of observing, during the experiment, finite element's nodes, N_2 is the number of time steps in the numerical experiment, U_{ij} are the reference temperatures generated with the reference heat transfer coefficient κ and T_{ij} are the temperatures obtained during the numerical experiment. The standard deviation value was used to present the variability of the temperature distribution, i.e. to determine the similarity of the obtained solutions to each other and the deviation from the mean value, taking into account the obtained value of the κ coefficient.

In order to execute a numerical experiment, reconstructing the thermal conductivity of the separating layer, both numerical (using TalyFEM) and optimisation (using ABC and ACO heuristic algorithms) calculations were carried out. The heuristic of the optimisation algorithms used requires the user to run the program several times. In our work, all calculations were run five times for each configuration. During each iteration, the temporary best values of the $\boldsymbol{\kappa}$ coefficient were obtained among all individuals participating in the search for a food source. Finally, the reconstructed coefficient was the mean value of the best values obtained during each program run. The best heat transfer coefficient is the one that is closest to the reference coefficient, and its functional has the lowest value. The general idea of our software is presented in Fig. 1.

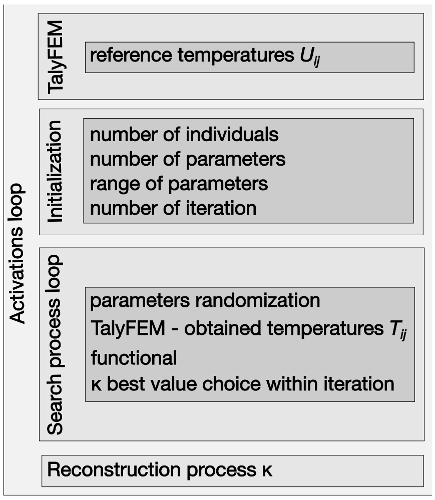


Fig. 1. Scheme of the problem solution

4. RESULTS

The considered geometry and the finite element mesh are presented in Fig. 2.

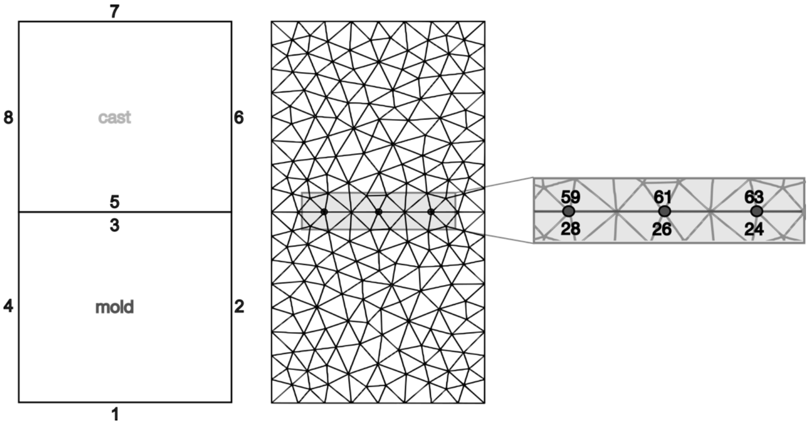


Fig. 2. Geometry and finite element mesh with denoted nodes

However, only one was chosen at a random density of finite element mesh because the differences in the results for all cases were minor. Detailed observations of the temperature course in time and the determined heat transfer coefficient κ were carried

DOI 10.2478/ama-2022-0039

out for three pairs of nodes. However, only one middle pair of nodes 26–61 (the first number of the node in a pair belongs to the mould, the other to the casting) was chosen to represent experiment results. The nodes were chosen symmetrically in sensitive (from the point of view of the macroscale) places. The nodes at the interface between the cast and the casting mould have the same spatial coordinates, making finite element meshes easier to automatically generate and separate, which simulates a layer separating two areas with completely different material properties. The mesh was composed of 360 finite triangular elements (214 nodes).

The numerical experiment for the Al-2%Cu alloy and parameter κ from the value range 900–1,500 $\left[\frac{W}{m^2K}\right]$ was carried out, and the reference temperatures U_{ij} were obtained for κ = 100$\left[\frac{W}{m^2K}\right]$. The initial temperatures were T_0 = 960 K for the cast and T_0 = 590 K for the casting mould. Material properties are shown in Tab. 1.

Tab. 1. Material properties to cast and casting mould.

Property	Symbol	Cast	Mould
Density	$\rho, \frac{\mathrm{kg}}{\mathrm{m}^3}$	2,824	7,500
Specific heat	$c, \frac{\mathrm{J}}{\mathrm{kgK}}$	1,077	620
Heat transfer coefficient	$\lambda, \frac{\mathrm{W}}{\mathrm{mK}}$	262	40

The boundary conditions are shown in Fig. 3. In the Newton boundary condition (the third kind of boundary condition), convective heat exchange with the environment was established, assuming that the ambient temperature is 300 K, and the heat exchange coefficient with the environment is 100 $\left[\frac{W}{m^2K}\right]$.

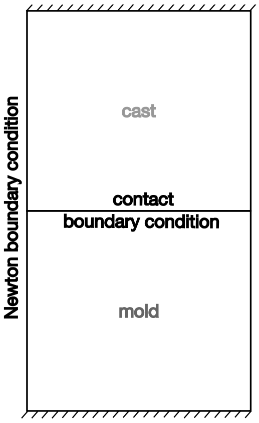


Fig. 3. Boundary conditions

Calculations were performed for the presented finite element mesh for the ABC and ACO algorithms for a population of 20, 40 and 60 bees/ants. A characteristic feature of heuristic algorithms is that they must be run more than once to give the possibility of narrowing down the search area and obtaining reliable calculation results. In this study, the algorithms were run five times in each case. Moreover, in each case, 0%, 1%, 2% and 5% noise of the reference temperatures U_{ij} were also considered. The following part of the article presents the results obtained from a numerical experiment for a given configuration presented in Section 4.

The tables presented in the article contain the results of the calculations (κ represents heat transfer coefficient, σ standard deviation and J the value of the minimised functional) using the ABC and ACO algorithms at 0%, 1%, 2% and 5% noise for 2, 6 and 12 iterations for finite element meshes with 214 nodes. Tab. 2 shows the calculations for 20 individuals (bees/ants) of the population, whereas Tab. 3 shows the calculations for 60 individuals' selected swarming algorithms. After analysing the results presented in the tables, the following conclusions can be drawn:

- the errors in the reconstruction of the κ coefficient are minor and do not exceed a few percent,
- even for disturbed input data, the errors in the reconstruction of the κ coefficient do not exceed the amount of the noise introduced,
- the values of standard deviations decrease for each of the algorithms (grey colour in Figs. 4 and 5) corresponding to an increase in the number of iterations,
- similar to the size of the noise increases, the value of the minimised functional increases, but the values of standard deviations decrease for each of the algorithms,
- a small number of iterations results in a more significant discrepancy between the obtained κ coefficient values and thus a more significant standard deviation,
- regardless of the perturbation value introduced, increasing the number of iterations of the algorithm execution does not significantly reduce the functional value, i.e. it does not significantly improve the reconstructed value of the κ coefficient. However, one could say the best representation of the κ coefficient was obtained for 12 iterations for each noise case.

Summarising, it can be generalised that the error in determining κ coefficient was smaller than the percentage variation in the reference temperature value in all cases. There are no significant differences between the selected algorithms, but the obtained standard deviation and functional results favour the ACO algorithm.

Tab. 2. Calculations for the ABC and ACO algorithm (20 bees/ants) for the κ coefficient with 0%, 1%, 2% and 5% noise of the reference temperature; σ is the standard deviation value expressed as a percentage; J is the functional value

Noise	Iterations	κ		σ %		J	
		ABC	ACO	ABC	ACO	ABC	ACO
0%	2	1,005.42	1,002.02	1.720797	0.315057	0.005593	0.001838
	6	999.971	999.982	0.191069	0.003093	0.000811	0.000017
	12	1,000.02	1,000.00	0.021884	0.000049	0.000107	0.000000
1%	2	1,001.568	999.738	1.511249	0.229253	0.159742	0.159514
	6	994.17	997.505	0.680394	0.010121	0.159566	0.159503

DOI 10.2478/ama-2022-0039

	12	997.594	997.65	0.023884	0.011118	0.159503	0.159503
2%	2	996.772	998.459	0.357987	0.405015	0.319783	0.319784
	6	996.797	997.88	0.197866	0.003302	0.319779	0.319776
	12	997.873	998.037	0.004802	0.011228	0.319776	0.319776
5%	2	1,002.746	1,007.952	0.388378	0.446690	0.806145	0.806146
	6	1,005.352	1,005.147	0.106457	0.022512	0.806141	0.806141
	12	1,005.287	1,005.061	0.031364	0.007003	0.806141	0.806141

ABC, artificial bee colony; ACO, ant colony optimisation

Tab. 3. Calculations for the ABC and ACO algorithm (60 bees/ants) for the κ coefficient with 0%, 1%, 2% and 5% noise of the reference temperature; σ is the standard deviation value expressed as a percentage; J is the functional value

Noise	Iterations	κ		σ %		J	
		ABC	ACO	ABC	ACO	ABC	ACO
0%	2	999.772	1,000.119	0.207614	0.043968	0.000982	0.000185
	6	1,000.254	1000.00	0.073764	0.000043	0.000347	0.000000
	12	1000.025	1,000.00	0.009152	0.000000	0.000041	0.000000
1%	2	998.251	998.286	0.6631	0.000064	0.159548	0.159504
	6	996.963	997.609	0.058549	0.000011	0.159504	0.159503
	12	997.649	997.561	0.010208	0.000010	0.159503	0.159503
2%	2	999.058	997.718	0.139638	0.121819	0.319777	0.319777
	6	998.001	997.891	0.012700	0.010942	0.319776	0.319776
	12	998.037	998.008	0.014075	0.014957	0.319776	0.319776
5%	2	1,003.347	1,005.048	0.246709	0.039688	0.806143	0.806141
	6	1,005.028	1,005.061	0.041702	0.005901	0.806141	0.806141
	12	1,005.075	1,005.050	0.019215	0.007564	0.806141	0.806141

ABC, artificial bee colony; ACO, ant colony optimisation

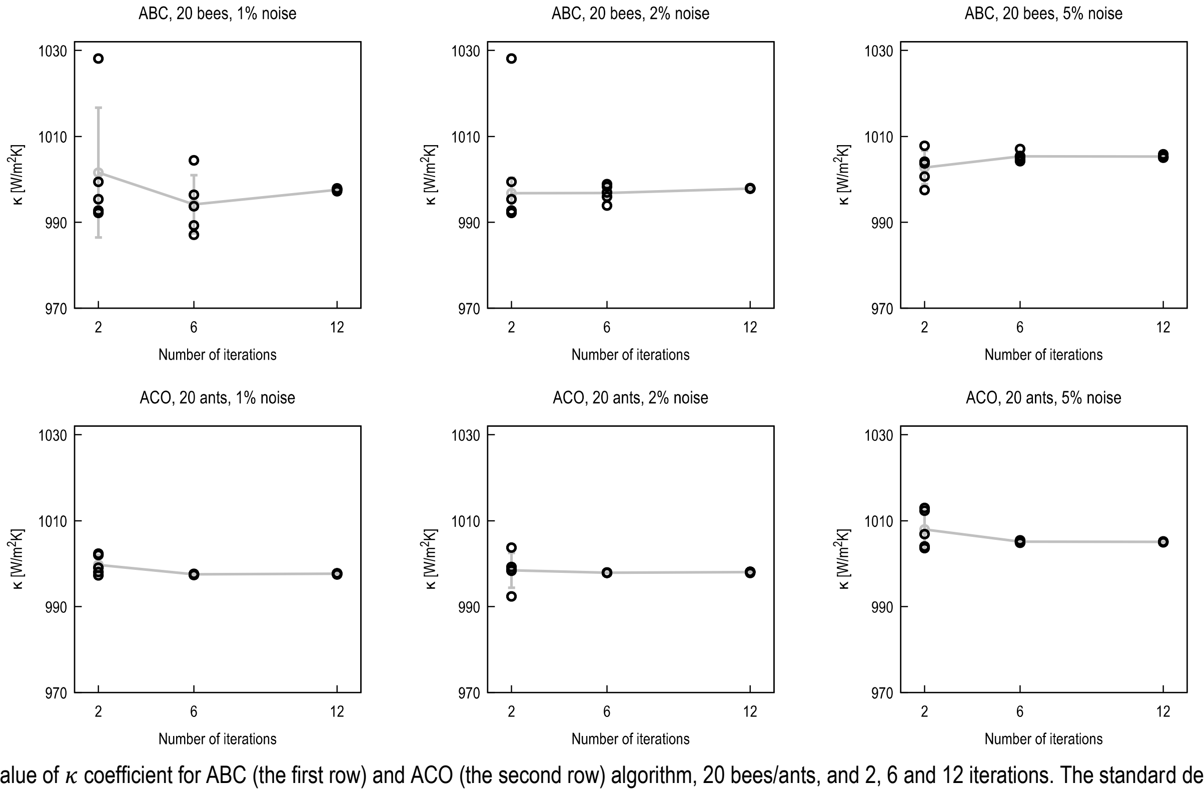


Fig. 4. Value of κ coefficient for ABC (the first row) and ACO (the second row) algorithm, 20 bees/ants, and 2, 6 and 12 iterations. The standard deviation and mean values of the κ coefficient are depicted by the grey color of the line and points. The exact value of the κ coefficient is 1,000.
ABC, artificial bee colony; ACO, ant colony optimisation

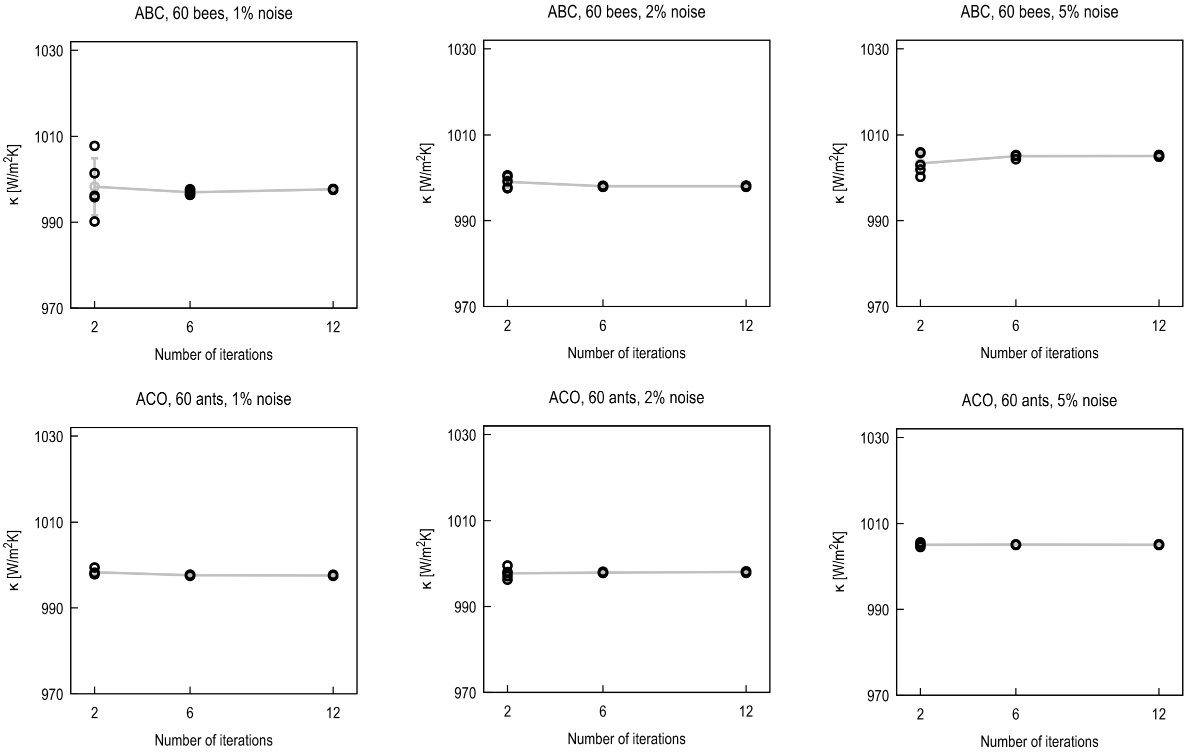
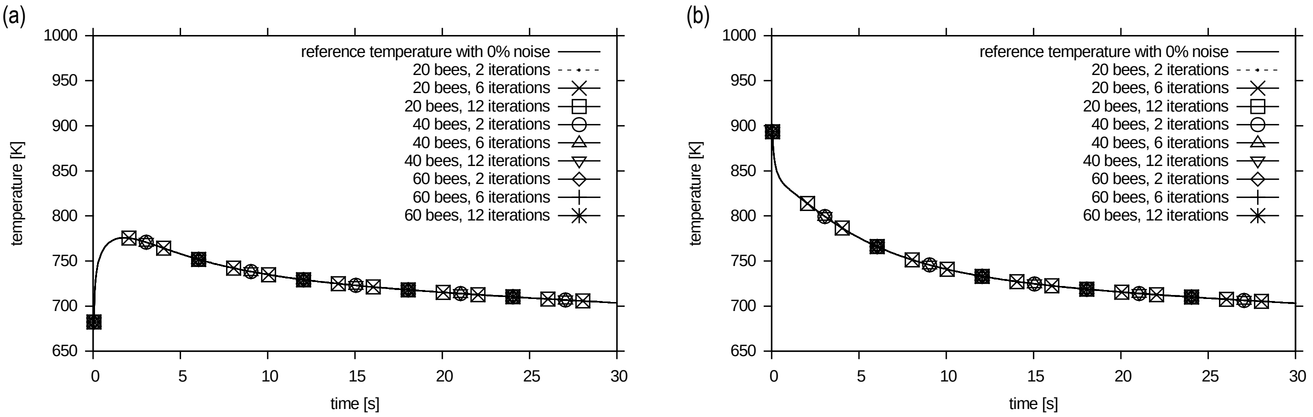


Fig. 5. Value of κ coefficient for ABC (the first row) and ACO (the second row) algorithm, 60 bees/ants, and 2, 6 and 12 iterations. The standard deviation and mean values of the κ coefficient are depicted by the grey colour of the line and points. The exact value of the κ coefficient is 1,000. ABC, artificial bee colony; ACO, ant colony optimisation

4.1. Analysis of the results obtained using the ABC algorithm

Figs. 6–9 show the temperature courses over time for the middle pair of nodes (26–61) in the finite element mesh of the studied geometry. In the case of ABC algorithm this figures visualise 0%, 1%, 2% and 5% disturbance for 20, 40 and 60 bees corresponding to 2, 6, and 12 iterations, respectively. The left panel tallies with the mould, and the right panel with the cast. The temperature distribution is physically correct in each case. The corresponding differences between the values of the temporary temperature T_{ij} and the reference temperature U_{ij} in the last row of each figure are shown. After about 20 s, it can be seen that the temperatures in the cast and the casting mould level out.

Moreover, there are unseen differences in the obtained temperatures, regardless of the ABC algorithm's iteration number. However, these differences do not exceed 1 K in any of the cases. It can be concluded from the graphs that there are no slight visible differences in the obtained temperatures, regardless of the number of individuals in the population and the number of iterations using the ABC algorithm. The time in the figures starts with the first time step, which is 0.05 s.



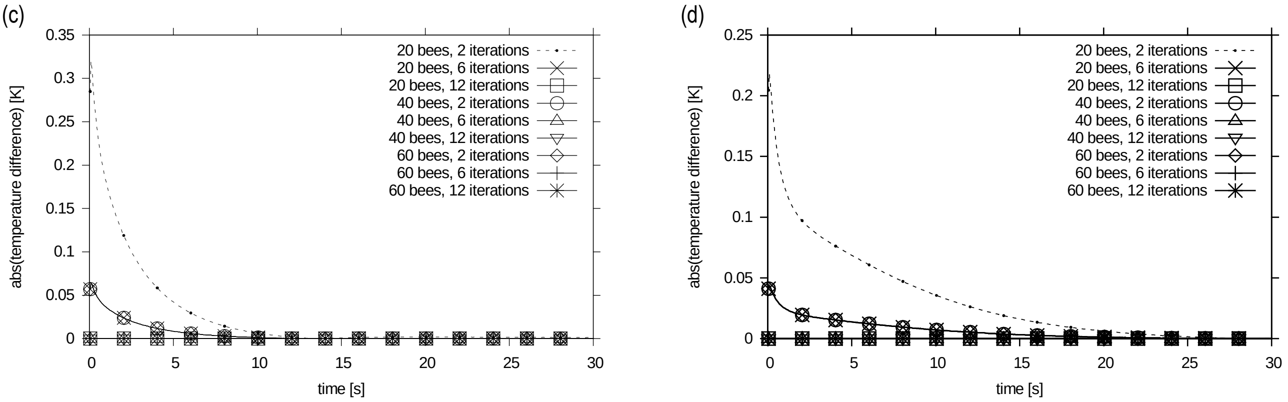


Fig. 6. Temperature over time distribution at 0% disturbance, for the ABC algorithm, with 20, 40 and 60 bees for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould, whereas the right panel for node no 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature for (a) the mould and (b) the cast are shown. The corresponding difference between values of temporary temperature T_{ij} and the reference temperature U_{ij} for (c) the mould and (d) the cast is depicted. ABC, artificial bee colony

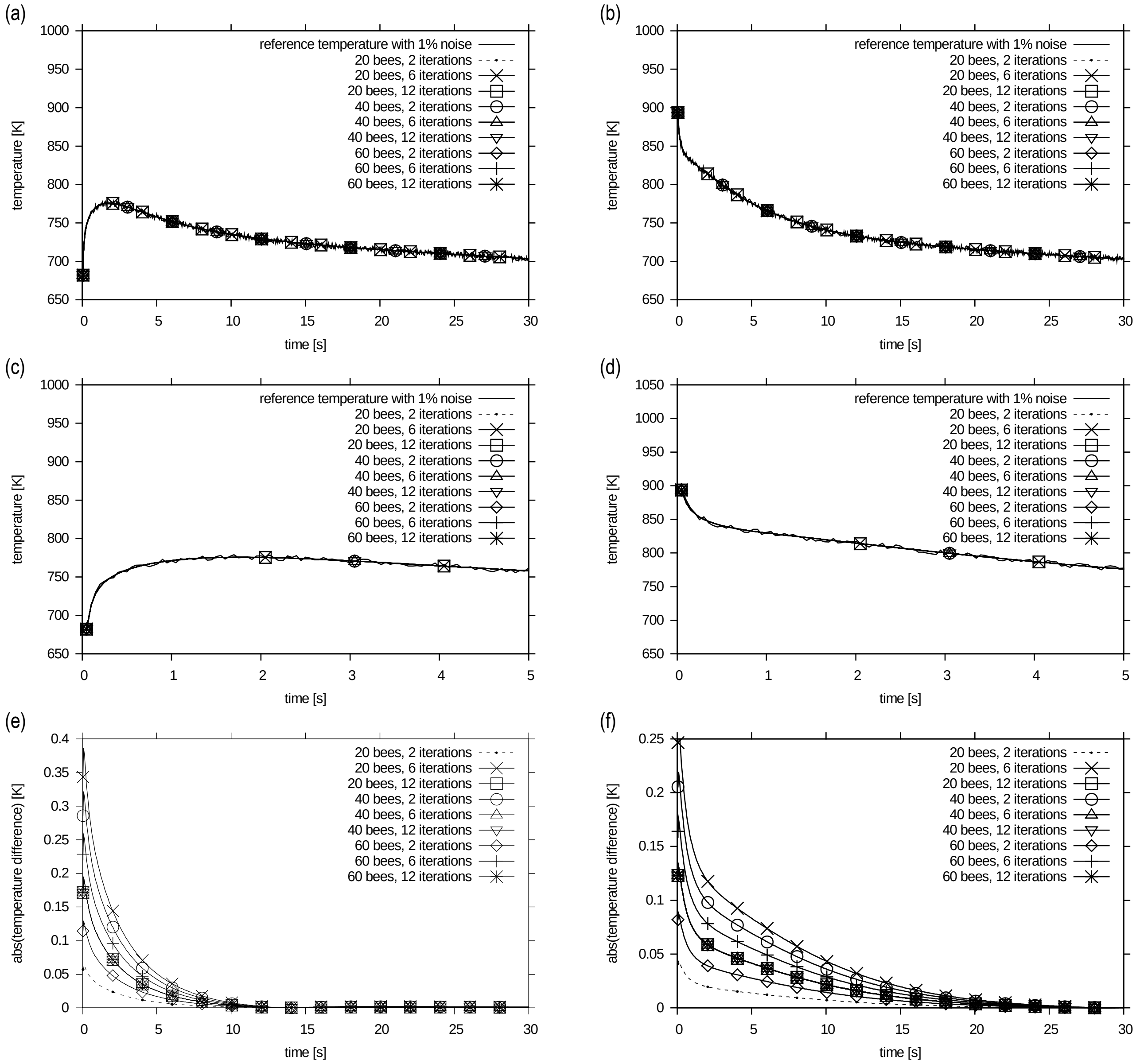


Fig. 7. Temperature over time distribution at 1% disturbance, for the ABC algorithm, with 20, 40 and 60 bees for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ABC, artificial bee colony.

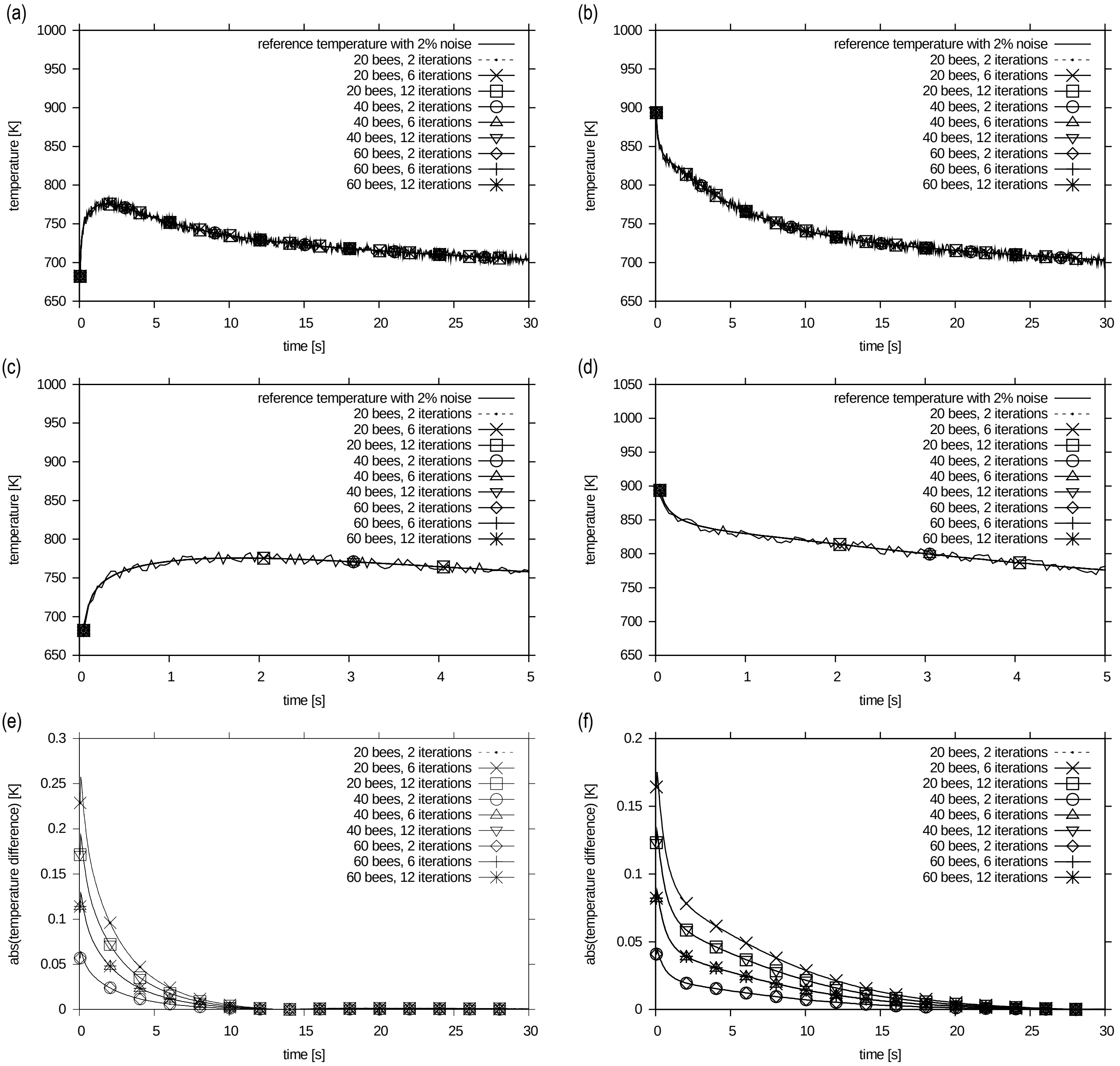
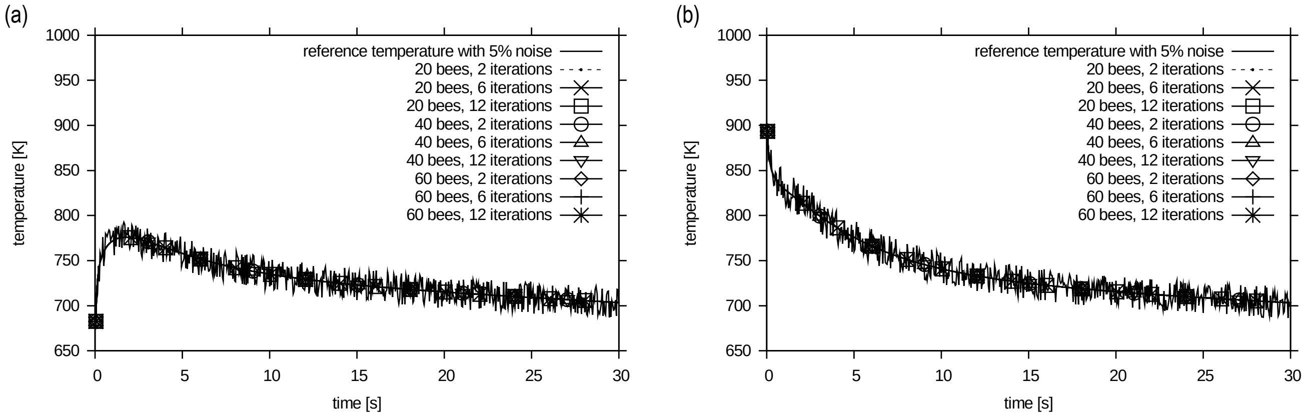


Fig. 8. Temperature over time distribution at 2% disturbance, for the ABC algorithm, with 20, 40 and 60 bees for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ABC, artificial bee colony



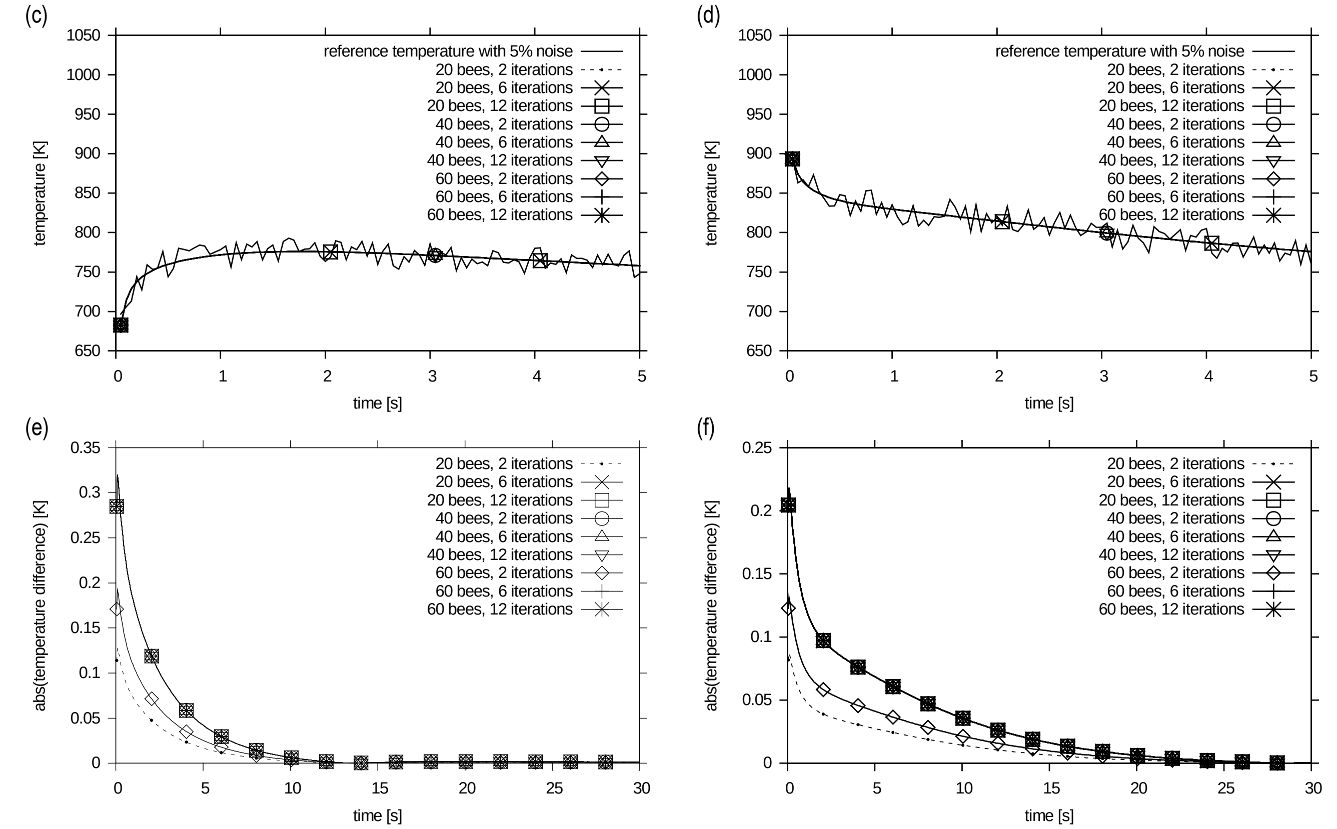
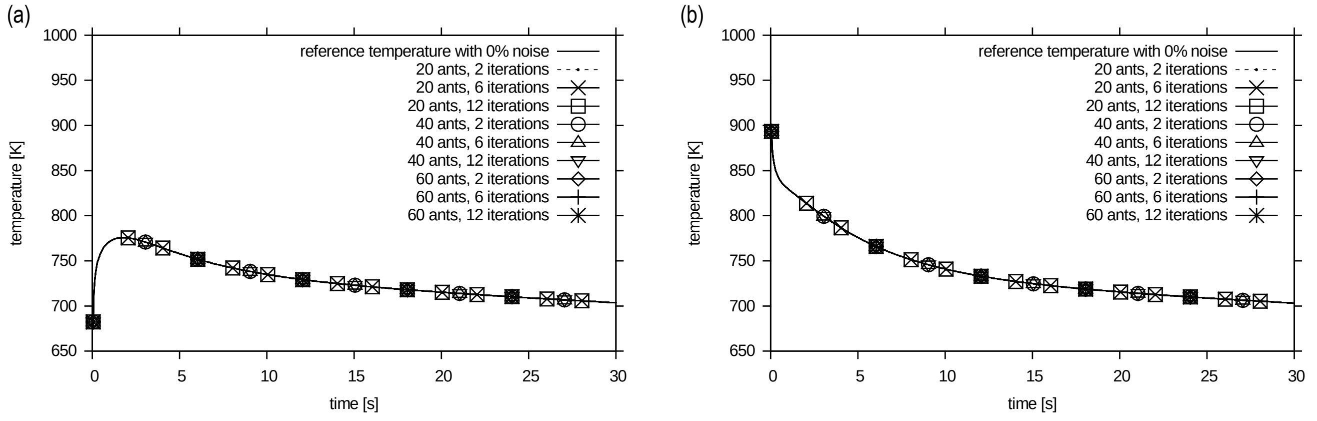


Fig. 9. Temperature over time distribution at 5% disturbance, for the ABC algorithm, with 20, 40 and 60 bees for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ABC, artificial bee colony

4.2. Analysis of the results obtained using the ACO algorithm

Figs. 10–13 show the temperature courses over time for the middle pair of nodes (26–61) in the finite element mesh of the studied geometry. In the case of ACO algorithm this figures visualise 0%, 1%, 2% and 5% disturbance for 20, 40 and 60 ants corresponding to 2, 6, and 12 iterations, respectively. The left panel tallies with the mould, and the right panel with the cast. The temperature distribution is physically correct in each case. The corresponding differences between values of the temporary temperature T_{ij} and the reference temperature U_{ij} in the last row of each figure are shown. After about 20 s, it can be seen that the temperatures in the cast and the casting mould level out.

Moreover, there are unseen differences in the obtained temperatures, regardless of the ACO algorithm's iteration number. However, these differences do not exceed 1 K in any of the cases. It can be concluded from the graphs that there are no slight visible differences in the obtained temperatures, regardless of the number of individuals in the population and the number of iterations using the ABC algorithm. The time in the figures starts with the first time step, which is 0.05 s.



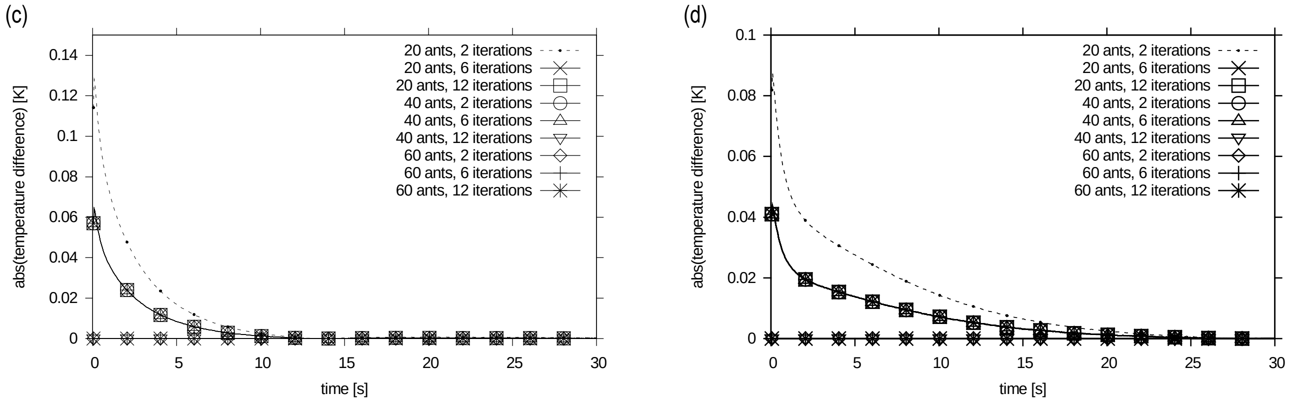


Fig. 10. Temperature over time distribution at 0% disturbance, for the ACO algorithm, with 20, 40 and 60 ants for 2, 6 and 12 iterations, respectively. Left panel depicts data obtained for node number 26 belonging to the mould and right panel for node number 61 belonging to the cast. Temperature vs. time courses with respect to the reference temperature for (a) the mould and (b) the cast. Corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (c) the mould and (d) the cast. ACO, ant colony optimisation

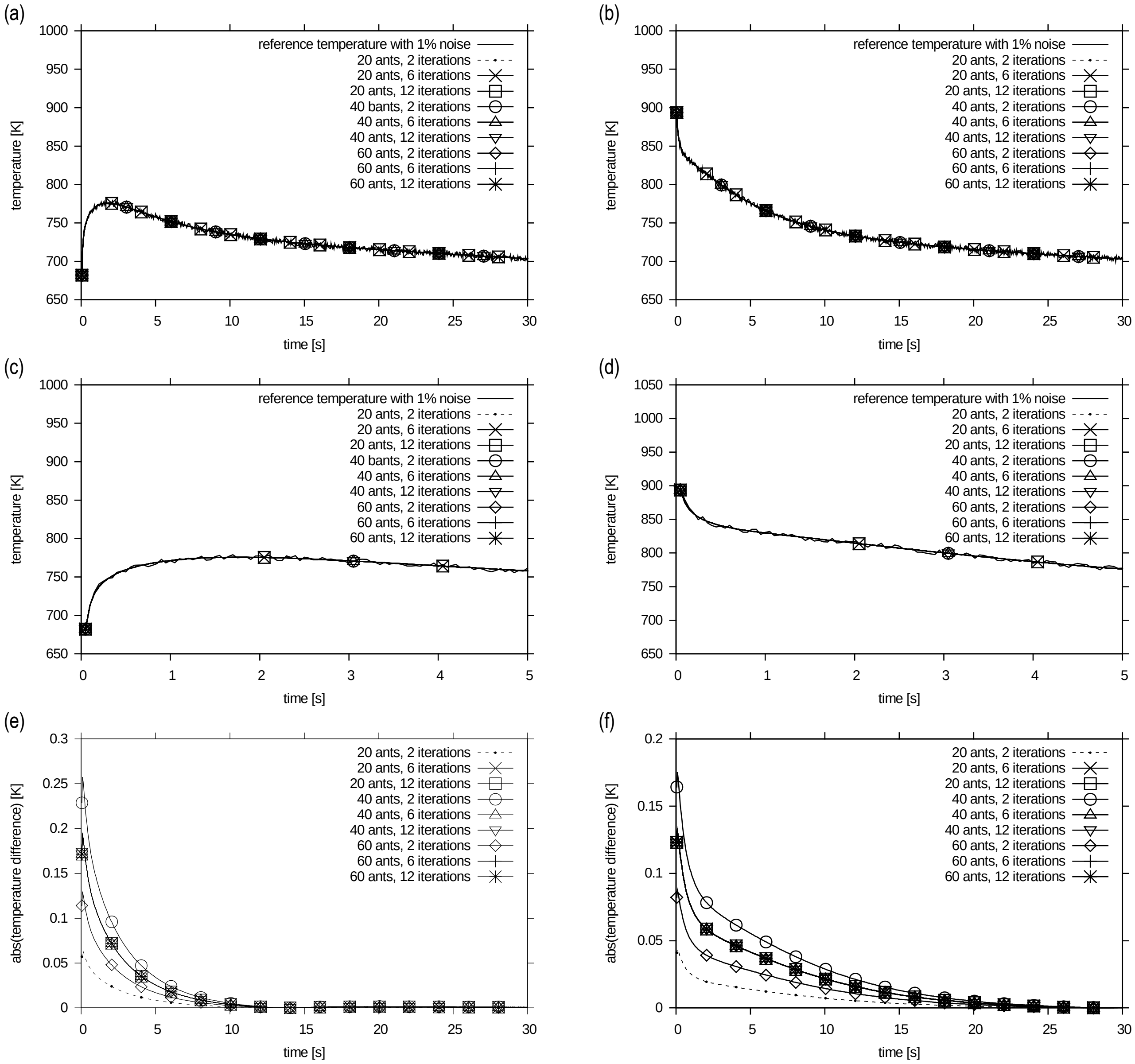


Fig. 11. Temperature over time distribution at 1% disturbance, for the ACO algorithm, with 20, 40 and 60 ants for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ACO, ant colony optimisation

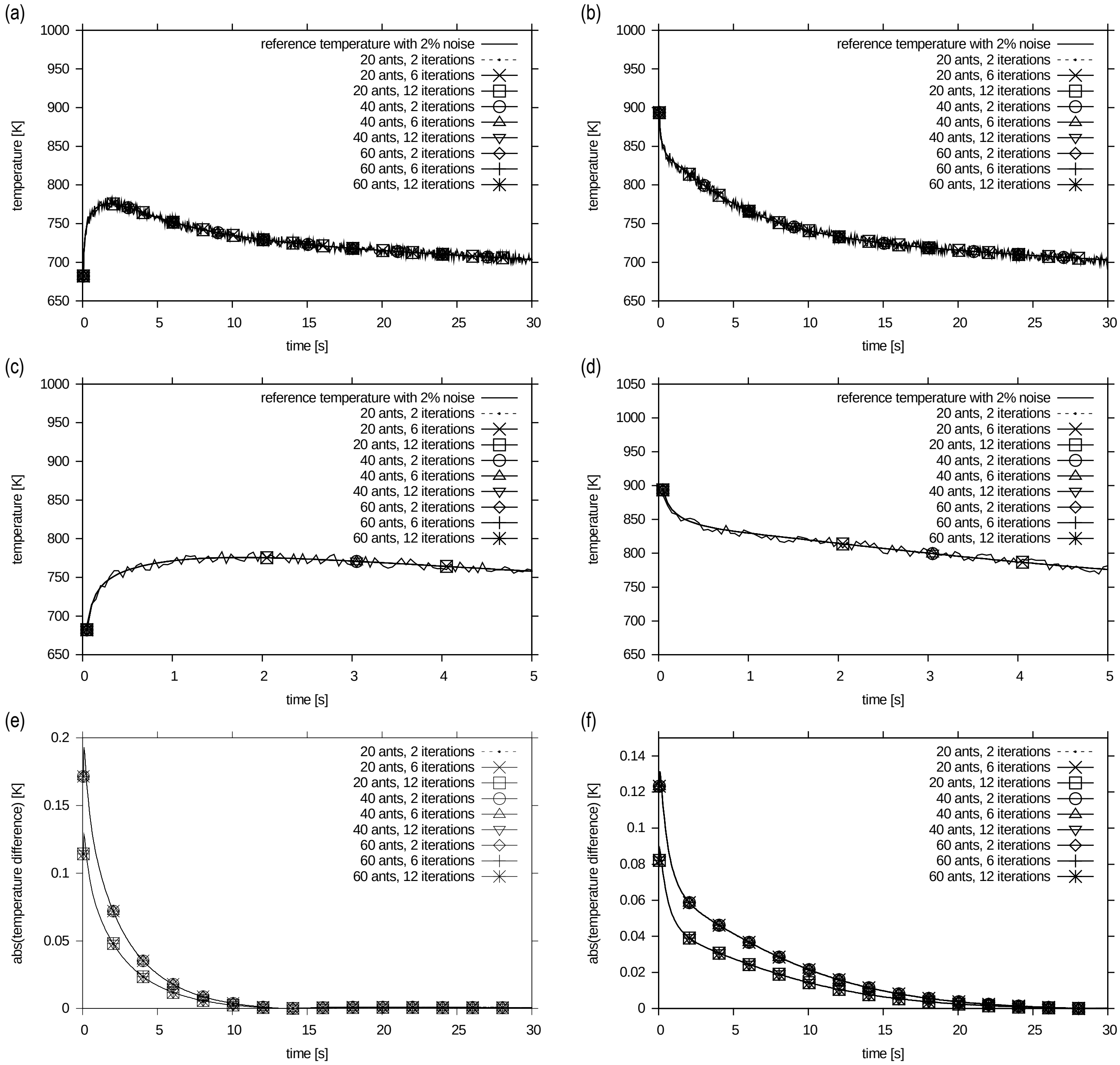
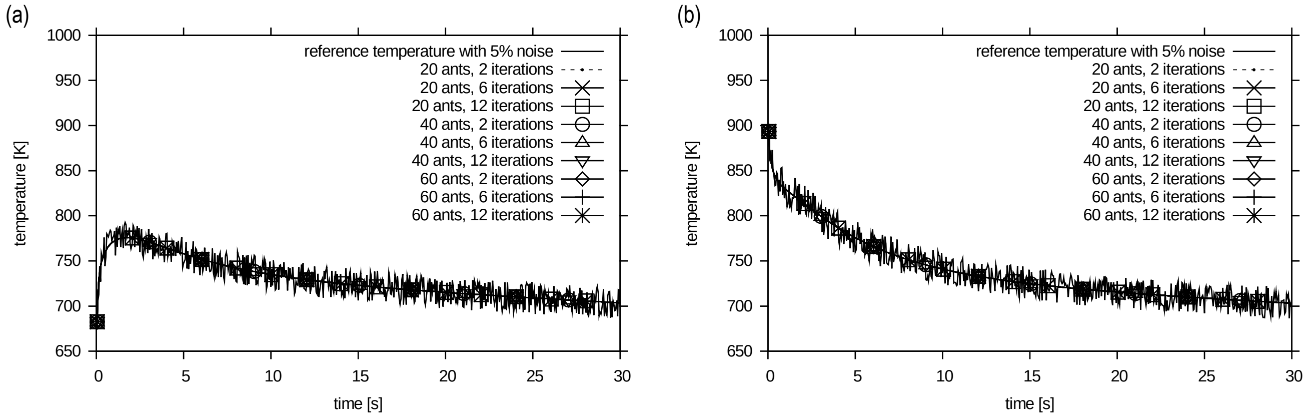


Fig. 12. Temperature over time distribution at 2% disturbance, for the ACO algorithm, with 20, 40 and 60 ants for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ACO, ant colony optimisation



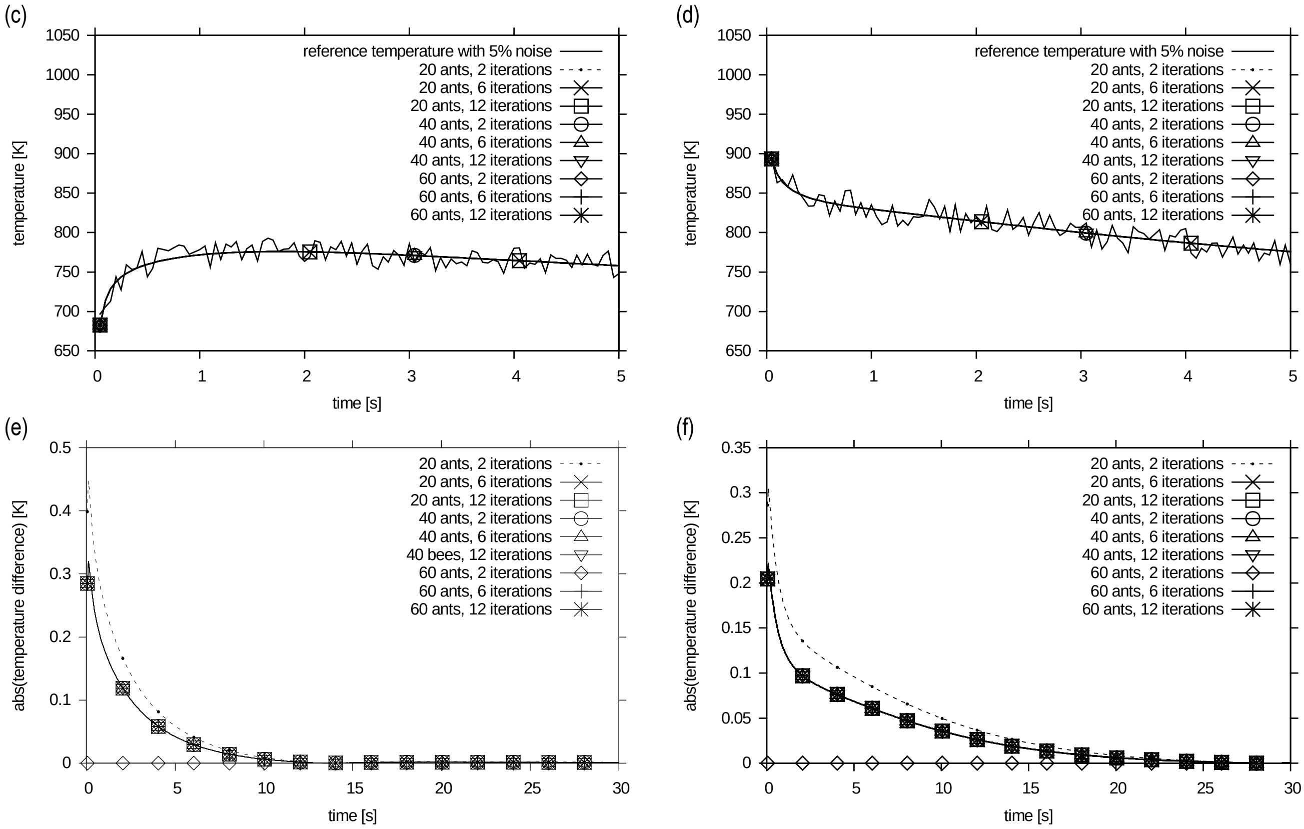
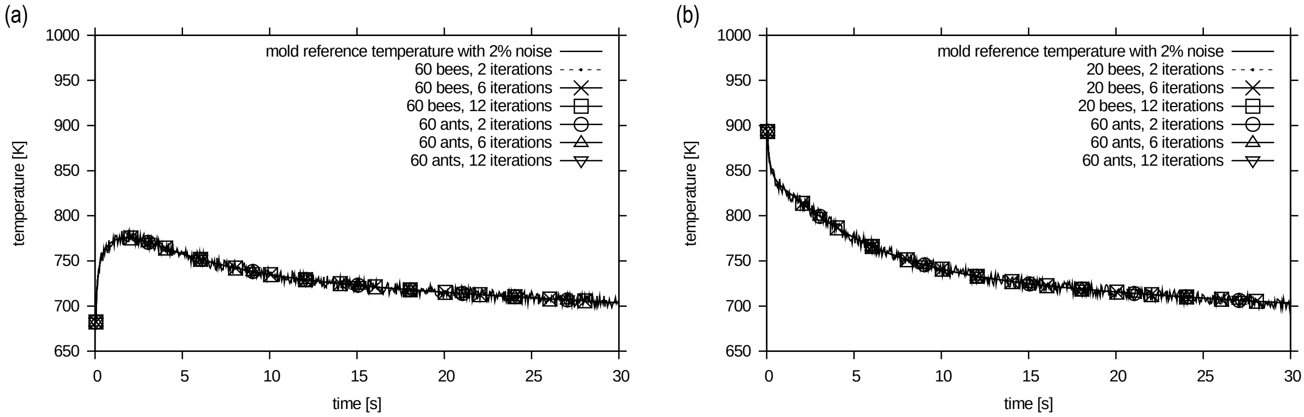


Fig. 13. Temperature over time distribution at 2% disturbance, for the ACO algorithm, with 20, 40 and 60 ants for 2, 6 and 12 iterations, respectively. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ACO, ant colony optimisation

4.3. Comparative analysis of the results for the ABC and ACO algorithms

Fig. 14 shows the temperature courses over time for the middle pair of nodes (mould-cast) in the finite element mesh of the studied geometry with a 2% disturbance of the temperature reference value for the second, sixth, and twelfth iterations and 60 bees/ants in ABC/ACO algorithm. Similar to the previous figures, after about 20 s, it can be seen that the temperatures in the cast and the casting mould level out. Moreover, the graph shows tiny differences (practically unseen) in the obtained temperatures between the ABC and the ACO algorithms. However, these differences do not exceed 0.2 K.

Both the ABC and the ACO algorithms meet the expectations for the correct determination of the κ coefficient and the correct course of temperature over time, consistent with the physics of the phenomenon.



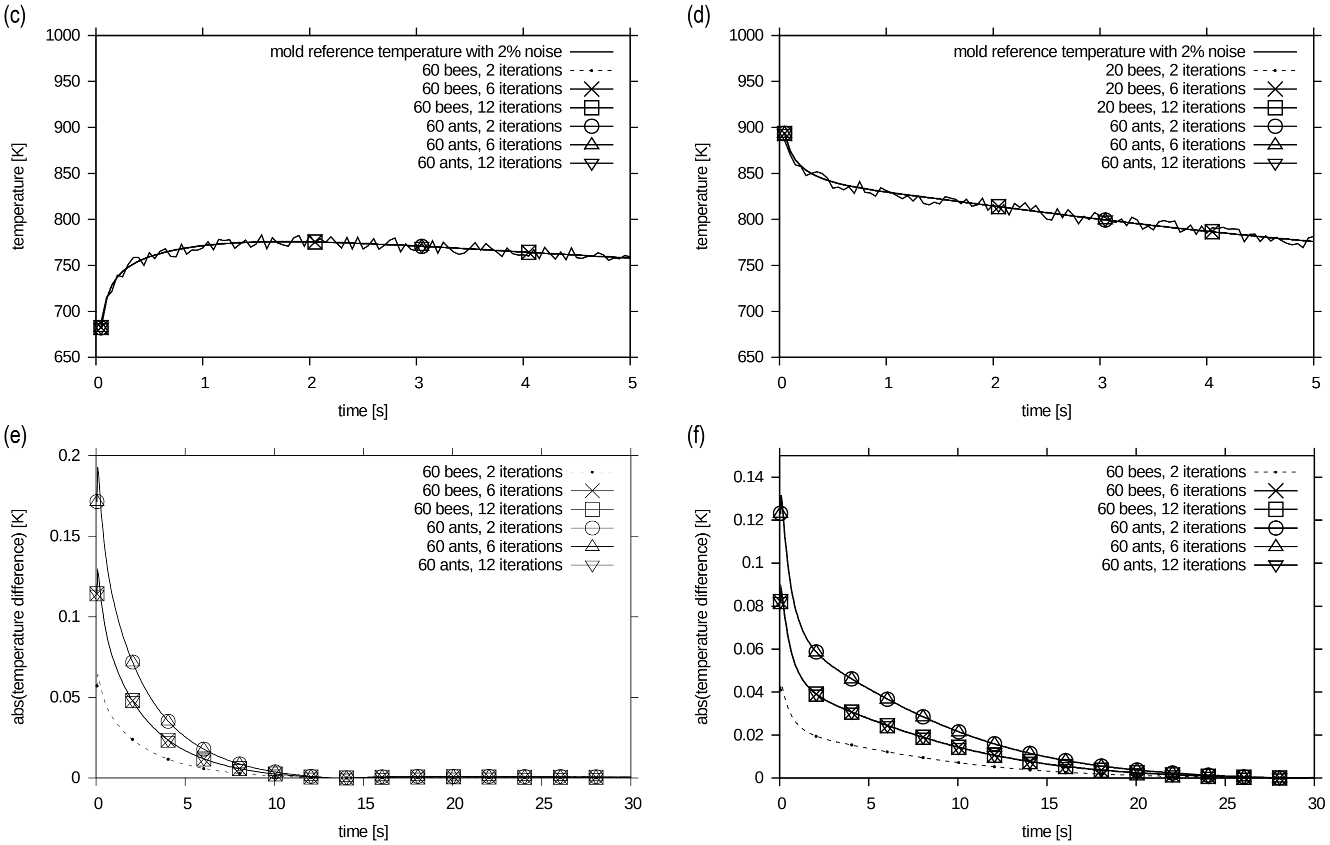


Fig. 14. Temperature over time distribution at 2% disturbance, for the ABC and ACO algorithms, with 60 bees/ants and for 2, 6 and 12 iterations. The left panel depicts data obtained for node number 26 belonging to the mould and the right panel for node number 61 belonging to the cast. Temperature vs. time courses concerning the reference temperature in (a) the mould and (b) the cast for the entire time of the numerical experiment, whereas (c) and (d) for the first 5 s, are shown. Furthermore, the corresponding difference between values of the temporary temperature T_{ij} and the reference temperature U_{ij} for (e) the mould and (f) the cast is depicted. ABC, artificial bee colony; ACO, ant colony optimisation

The possibility of using AI algorithms in problems in the field of thermomechanics was presented. In reconstructing the heat transfer coefficient of the layer separating the cast and the casting mould, no significant difference was observed between the ABC algorithm and the ACO algorithm. The graphs of temperature variability over time show a very good representation of reality for both swarming algorithms. After analysis of the results obtained with the ABC and ACO algorithms, it can be concluded that: for simple geometry and the same number of individuals, slight differences in the obtained values of the κ coefficient and the values of the minimised functional may speak in favour of the ant colony algorithm. The conducted research has shown that both the ABC and ACO algorithms are promising tools that can be successfully used to determine the value of the thermal conductivity coefficient. The errors in the reconstruction of the κ coefficient and the standard deviation are very similar between the algorithms. Equally important, they are smaller, or at worst, comparable to the disturbance of the input data. It has been shown that both algorithms select the desired coefficient at a satisfactory level. For each algorithm, the obtained temperature results were very similar or identical to the values assumed as the standard ones.

5. SUMMARY

The investigated problem consisted of reconstructing the heat transfer coefficient at the interface between the cast and the casting mould based on a numerical experiment using artificial bee and ant colonies algorithms for optimisation. The presented solution has been described and tested to assess its stability and the accuracy of the results obtained. The numerical experiment demonstrated an excellent reconstruction of the sought coefficient as well as perfect and physically similar consistency of temperature distribution over time. In each case of the input data and frequency of measurements taken, the errors in the mapping of the heat conduction coefficient and temperature were lower or comparable with the input data errors. The proposed solution is stable, and the obtained results are satisfactory, which bodes well for the future use of our research to recreate the experimental conditions for the solidification and cooling of non-ferrous metal alloy castings.

It is important to mention the possibility for recreation of the heat conduction coefficient of the layer separating the mould and the casting in the form of several intervals, i.e., by obtaining several numerical values; this is an arena of study that has not been explored in the present article, but it would be interesting to explore it as part of future work. Also, the possibility of obtaining a continuous variation in the value of the heat transfer coefficient through the layer separating the mould and the casting can be mentioned as another potential direction for further research. The suggested study of the impact of the distribution of nodes in which the error value was calculated is also a possible direction for further research.

REFERENCES

1. Gosselin L, Tye-Gingras M, Mathieu-Potvin F. Review of utilization of genetic algorithms in heat transfer problems. International Journal of Heat and Mass Transfer. 2009; 52(9-10):2169-2188.
2. Kot V. Solution of the classical Stefan problem: Neumann condition. Journal of Engineering Physics and Thermophysics. 2017; 90(4): 889-917.
3. Chen J, Yu W, Tian J, Chen L, Zhou Z. Image contrast enhancement using an artificial bee colony algorithm. Swarm and Evolutionary Computation. 2018; 38:287-294.
4. Zhao X, Xuan D, Zhao K, Li Z. Elman neural network using ant colony optimization algorithm for estimating of state of charge of lithium-ion battery. Journal of Energy Storage. 2020; 32:101789.
5. Karaboga D, Gorkemli B, Ozturk C, Karaboga N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. Artificial Intelligence Review. 2014; 42:21-57.
6. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report. Kayseri/Türkiye: Erciyes University, Engineering Faculty, Computer Engineering Department; 2005. Report No.: TR-06.
7. Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing. 2008; 8(1):687-697.
8. Hetmaniok E, Słota D, Zielonka A. Artificial Bee Colo-ny Algorithm Used for Reconstructing the Heat Flux Density in the Solidification Process. In International Conference on Artificial Intel-ligence and Soft Computing; 2014; 363–372.
9. Hetmaniok E, Słota D, Zielonka A, Wituła R. Comparison of ABC and ACO Algorithms Applied for Solving the Inverse Heat Conduction Problem. In International Symposium on Swarm Intelligence and Differential Evolution; 2012; 249–257.
10. Hetmaniok E, Słota D, Zielonka A. Restoration of the cooling conditions in a three-dimensional continuous casting process using AI algorithms. Applied Mathematical Modelling. 2015; 39(16): 4794-4807.
11. Zielonka A, Hetmaniok E, Słota D. Inverse alloy solidification problem including the material phenomenon solved by using the bee algorithm. International Communications in Heat and Mass Transfer. 2017; 87:295-301.
12. Grzymkowski R, Hetmaniok E, Słota D, Zielonka A. Application of the Ant Colony Optimization Algorithm in Solving the Inverse Stefan Problem. In Metal Forming; 2012; 1287-1290.
13. Hetmaniok E, Słota D, Zielonka A. Application of the Swarm Intelligence Algorithm for Investigating the Inverse Continuous Casting Problem. Contemporary Challenges and Solutions in Applied Artificial Intelligence. 2013; 489: 157–162.
14. Matsevityi YM, Alekhina SV, Borukhov VT. Identification of the thermal conductivity coefficient for quasi-stationary two-dimensional heat conduction equations. Journal of Engineering Physics and Thermophysics. 2017; 90(6):1295-1301.
15. Tereshko V, Loengarov A. Collective decision-making in honey bee foraging dynamics. Computing and Information Systems. 2005; 9: 1-7.
16. Colorni A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies. In Proceedings of the European Conference on Artificial Life; 1991; 134-142.
17. Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics). 1996; 26(1):29-41.
18. Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99; 1999;1470-1477.
19. Geuzaine C, Remacle JF. GMSH: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering. 2009; 79(11):1309-1331.
20. Dyja R, Grosser A. Oblicznia równoległe w symulacji krzepnięcia wykorzystującej model pośredni narstania fazy stałej. Modelowanie Inżynierskie. 2015; 24(55):21-26.
21. Dyja R, Gawronska E, Grosser A, Jeruszka P, Sczygiol N. Estimate the Impact of Different Heat Capacity Approximation Methods on the Numerical Results During Computer Simulation of Solidification. Engineering Letters. 2016; 24(2):237-245.
22. Kodali HK, Ganapathysubramanian B. A computational framework to investigate charge transport in heterogeneous organic photovoltaic devices. Computer Methods in Applied Mechanics and Engineering. 2012; 247:113-129.
23. Balay S, Gropp WD, McInnes LC, Smith BF. Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In Arge E,BAM,LHP. Modern Software Tools for Scientific Computing. Boston. 1997;163–202.
24. Dyja R. Comparison of Results from In-House Solidification Convection Model with Standard Benchmark. Acta Physica Polonica. 2021; 139(5):525-528.

Elżbieta Gawrońska: https://orcid.org/0000-0003-1060-3641

Robert Dyja: https://orcid.org/0000-0001-6941-4728

Maria Zych: https://orcid.org/0000-0002-0406-6053

Grzegorz Domek: https://orcid.org/0000-0003-3566-9110