Journal of Applied Mathematics and Computational Mechanics 2022, 21(4), 107-118
www.amcm.pcz.pl p-ISSN 2299-9965
DOI: 10.17512/jamcm.2022.4.10 e-ISSN 2353-0588

IMPACT OF INPUT PARAMETERS ON NUMERICAL CALCULATIONS OPTIMIZED BY SWARMING ALGORITHMS DURING COMPUTER SIMULATIONS OF THE HEAT CONDUCTION PHENOMENON

Maria Zych[1], Elzbieta Gawronska[1], Robert Dyja[1], Piotr Krawiec[2]

[1] *Faculty of Mechanical Engineering and Computer Science, Czestochowa University of Technology Czestochowa, Poland*
[2] *Faculty of Mechanical Engineering, Poznan University of Technology Poznan, Poland*
maria.zych@icis.pcz.pl, elzbieta.gawronska@icis.pcz.pl, robert.dyja@icis.pcz.pl piotr.krawiec@put.poznan.pl

Received: 8 October 2022; Accepted: 29 November 2022

Abstract. The article presents the application of swarming algorithms in heat conduction, taking into account the continuity of the boundary condition (type IV). The influence of the input parameters of the bee and ant algorithm and tessellation on the selection of the heat conduction coefficient between the casting mold and the casting in computer simulations was presented. The results were compared for two different finite element grids, a different number of individuals, and a different number of iterations. The study also considered the magnitude of the reference temperature disturbance as the input temperature for numerical calculations. The analysis showed that the relative error of reproducing the value of the thermal conductivity coefficient in the continuity condition did not exceed 1.5% of the reference value of this coefficient.

MSC 2010: *83A99, 65L50, 68V99, 68T99*
Keywords: *computational mechanics, thermomechanics, swarming algorithms, heat conduction*

1. Introduction

Artificial Intelligence (AI), with the help of computation, seeks to solve imposed issues effectively and non-algorithmizable based on modeling knowledge. An example of such an issue is image processing, such as face recognition or handwriting. Human intelligence can be a point of reference here if the issues of artificial intelligence are considered. The essential functions that make up human intelligence are the processes of learning and the practical use of knowledge, association, generalization, and cognitive abilities. The most important learning processes include memorization, achievement of goals, ability to interact, the definition of conclusions, analytical fluency, and conceptual and abstract thinking. Intelligent machines created by man can be programmed so that the elements mentioned earlier of human intelligence are imitated only in a narrow range. Many models of such machines are described in the literature [1–3].

Karaboga et al. [4] proposed a new bee swarm model called the Artificial Bee Colony algorithm (ABC). It is modeled on the intelligent behavior of honey bees during their food acquisition. In nature, a swarm of these bees is considered the most intelligent. This model consists of three main elements: food sources, unemployed bees, and employed bees. Bees have developed various techniques in searching for food, such as waggle dance, to improve communication with other bees about the location of food sources.

A completely different issue is the problem of selecting the shortest route between two different points in the city at a certain distance. In this case, the implemented Ant Colony Optimization (ACO) algorithm was used, which takes full advantage of the computational capabilities of multiprocessor and distributed systems. The experiment carried out was to demonstrate the superiority of using the algorithm over a system of traditional navigation [5, 6].

Karaboga in work [7] presents a new approach to solve the inverse heat conduction problem in estimating an unknown heat source. He formulated the physical heat transfer problem as an optimization problem. The modified genetic algorithm falls into the class of heuristic algorithms, which search for possible solutions to find a solution better than the original one. Such a search is carried out by evolutionary mechanisms, as well as a natural selection, and was developed to solve the resulting optimization problem.

Swarm algorithms easily fit the size constraints of the solution space with no dependence on the number of variables. Beni et al. and Hackwood et al., in their works [8, 9], presented the concept of swarm intelligence. The inspiration for the development of these algorithms came from observations of biological systems such as flocks of birds, swarms of ants, colonies of worms, or just swarms of bees.

In the works of Hetmaniok et al. [10, 11], the procedure for solving the inverse thermal conductivity problem with a boundary condition of the third kind using selected swarm intelligence algorithms was realized. The solution to this problem involved identifying the thermal conductivity parameter and reconstructing the temperature distribution in the area. In the presented articles, the results of minimizing the functional determined the error of the approximate solution in the issue of heat conduction. Moreover, the results show that the proposed algorithm is an effective tool for solving this kind of inverse problem for different cases of the error of the input data, the distance of the control point from the boundary of the area, and the selection of parameters in algorithms of swarming.

The article presents the application of swarming algorithms in heat conduction simulations. The finite elements method (FEM) is the most applied method used for numerical calculation of many phenomena in computer simulations, for example, in thermomechanics, and many others [12, 13]. Therefore the authors use FEM in the numerical part of their research. In the next part of this paper, we present the influence of the input parameters of the bee and ant algorithms and tessellation on the reconstruction of the heat conduction coefficient between the casting mold and the casting in computer simulations. The results were compared for two different finite element grids, a different number of individuals, and a different number of iterations [14].

2. The mathematical model

2.1. Heat conduction

During the process of heating and cooling bodies, there is undetermined conduction of heat as long as the bodies strive to achieve temperature equilibrium with the environment in which they are located. The thermal exchange between parts of bodies that are in direct contact with each other is defined as heat conduction. The following formula defines the mathematical model of heat transfer by conduction:

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

where: ρ – density of the tested material $\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$, Q – capacity of internal heat sources $\left[\frac{\mathrm{W}}{\mathrm{m}^3}\right]$ (in this paper $Q = 0$ due to lack of such sources), ∇ – differential nabla operator, T – temperature [K], c – specific heat $\left[\frac{\mathrm{J}}{\mathrm{kgK}}\right]$, $\frac{\partial T}{\partial t}$ denotes the first derivative of temperature with respect to time.

The issue of transient heat conduction belongs to initial-boundary value problems, requiring the task of appropriate initial and boundary conditions. Initial conditions, called Cauchy conditions, give certain body temperature values at the initial moment $t_0 = 0$ s.

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

where $\mathbf{r}$ is the field vector at a given point. To determine the transient temperature distribution, the condition given by the formula (2) is necessary [15].

We distinguish four types of boundary conditions associated with complex heat transfer:

- A boundary condition of the first kind (Dirichlet), at the edge Γ_A of the area Ω temperature is set (T_z)

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

- The boundary condition of the second kind (von Neumann), at the edge Γ_B of area Ω the heat flux is known (q_z)

$$\Gamma_B : q = (q_z), \tag{4}$$

- A boundary condition of the third kind (Newton's or Robin's), at the edge Γ_C of the area Ω, heat exchange with the environment takes place:

$$\Gamma_C : q = \alpha(T - T_{env}), \tag{5}$$

where α is the heat transfer exchange with the environment, T is the body temperature at the edge Γ_C and T_{env} is the ambient temperature, q denotes the heat flux entering ($T < T_{env}$) into the area Ω or flowing out ($T > T_{env}$) from the area Ω.

- The boundary condition of the fourth kind (continuity condition), at the edge Γ_D between areas Ω_1 and Ω_2, heat flow occurs. Two cases are possible here:
  - ideal contact between areas
  - lack of ideal contact (coefficient κ is describing heat exchange through the separation layer)

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

where λ_p is the thermal conductivity coefficient of the separation layer of the protective covering, and δ is the thickness of that layer [16].

2.2. Swarm algorithms

Artificial Bee Colony and Ant Colony Optimization are classified as swarm algorithms. They are metaheuristic algorithms for solving various types of computational problems. They are used to optimize numerical problems and belong to the class of herd algorithms.

2.2.1. Bee algorithm

In the ABC algorithm, a colony of artificial bees consists of three groups of bees. The first half of the colony consists of employed bees and is one of the three groups. Another half of the colony includes a group of scout bees and a group of unstaffed bees. The artificial bee colony optimization algorithm assumes that each food source belongs to only one bee, meaning that the number of employed bees equals the number of food sources around each hive. Bees whose food source has been exhausted become unemployed bees. In the ABC algorithm, the position of the food source represents a possible solution to the optimization problem, and the food source's nectar content corresponds to the associated solution's quality (efficiency). At the first stage, ABC randomly generates an initial population P, the size of the number of food sources, solutions SN, where SN denotes the population size and the number of food source positions. Each solution x_i denotes the positions of the food source, $(i = 1,2,...,SN)$ is a vector of solutions of size D. In the algorithm, D denotes the number of optimization parameters. After initialization, determining the coordinates of the positions of food sources is subjected to multiple cycles $C = 1,2,...,C_{max}$, where C_{max} is the maximum number of cycles of execution of the algorithm. Cycles denote the update of the solution. The employed bees update the position (solution) changes according to local information (visual information) and test the amount of nectar (fitness value) of the new source (new solution). If the new nectar amount is higher than the previous cycle, the bee remembers the new position and forgets the previous one. Otherwise, it keeps the position of the previous one in its memory. Once the search process is complete, all employed bees share nectar information from different food sources and their position information with bees in the dancing area. The artificial non-employed bee selects a food source according to the probability value associated with that food source p_i, calculated according to the following formula:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \;\; i = 1, \ldots, P, \tag{7}$$

where fit_i is the value of the solution's efficiency i, which is proportional to the amount of nectar of the food source in the positions i. In this way, the bees that have worked, exchange their information with the individuals that observe them. In the next step, the coordinates of the food sources are updated according to the relation:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{8}$$

where v_{ij} is an update of the coordinates of the food sources, $k \in 1,2,...,SN$ and $j \in 1,2,...,D$ are randomly selected indices, such that $k \neq i$ and ϕ_{ij} is a random number from the interval [–1, 1] [17].

2.2.2. Ant algorithm

The ant algorithm is a technique designed mainly for problems of finding the best paths in a graph. Its inspiration comes from the world of ants, which can find the shortest route between the anthill and available food. As ants wander toward food, they choose a route at random, but as they return to the anthill, they leave a pheromone trail along their route, which, if left on the path, gradually evaporates if the path is not frequented. On a shorter route, evaporation is slower than on longer routes, so ants choose this route more readily than other routes. As a more favorable route is found, subsequent ants choose it by reinforcing the pheromone trail, known as the positive feedback phenomenon. In the ant algorithm, a colony of artificial individuals cooperates as they search for optimal solutions to complex combinatorial problems. There is an indirect interaction between ants and some form of gathering experience and using it in further exploration. Over time, the ants collectively develop a set of shortest paths leading them to their designated goals. It is called collective intelligence. There are several differences between real and artificial individuals. The first of these differences is that artificial ants move only within the boundaries set by the input graph, while real ants can take an arbitrary route. In the case of the ACO algorithm, the pheromone trace is related to the quality of the solution. It is also worth raising the question of how artificial ants find the desired solution. In ACO algorithm, each ant in each circuit finds a specific solution. The result of the entire program is the solution found by the best ant. If the route found by the artificial individual is better than the one generated so far, a pheromone trace is updated on the route so that subsequent ants are more likely to choose certain edges in the graph. The stronger the trace of the pheromone left, the more likely it is that a second ant will follow the trail of its predecessor. The factor influencing the strengthening of the pheromone trail is also the distance from the anthill to the foraging site (the path length in the graph).

All ants' passage routes occur according to the following rules. First, the nodes through which the ant k $(k = 1,\ldots,M)$ will pass are determined randomly, where M is the number of ants. Equation (9) defines the probability p_{ij} of choosing node j for an ant located in node i:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{i,j\in G^k}\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}, \quad i = 1,\ldots,D, \quad j = 1,\ldots,R, \tag{9}$$

where η represents the heuristic function, α, β are constants that determine the relative influence of pheromone and heuristic values on the ant's decision, G^k is a possible path for ant k to realize in the created graph, and D is still number of optimization parameters, R is a node in the graph. In the optimal layout, the value of this parameter is set, then the layout's operation is simulated, and the minimum quality index J_E is determined. Ant k leaves on the path of movement the amount of pheromone equal to $\Delta\tau_{ij}^{k}$. If the obtained path is better than the previous, this path is remembered in place of the previous best path of passage. The pheromone array is updated after all ants have traveled all paths, based on the formula:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{M}\Delta\tau_{ij}^{k}(t) + \rho\Delta\tau_{ij}^{best}(t), \tag{10}$$

where ρ is the evaporation coefficient $0 < \tau \leqslant 1$, and M is the number of ants.

Pheromone pairing is added to the algorithm to avoid unlimited growth of pheromone traces. In the first cycle for each ant, the nodes on the transition paths are randomized using the roulette wheel method, in which the probability calculated from the formula (9) is included. After the first ants' passage (on the path from the anthill to the food source), the best quality path index is found. Based on the defined quality index J_E, which obtained the highest rating, a modification of the ant's passage path is made. For the best transition path, new nodes in each layer are determined randomly. The nodes in each layer move closer to the path node with the best quality index. According to the formula (10), pheromone array is modified in each cycle of calculation. Finally, it is returned to the calculation of probability considering the determined pheromone array τ and proceeds to the next calculation cycle.

3. Assumptions for the research

The article deals with a topic that requires combining two separate fields of knowledge: thermomechanics and computer science. The heat conduction equation was chosen from the scope of the first field, while two artificial intelligence algorithms modeled by nature were selected from the second field. The physical process of heat conduction has been analyzed, taking into account the boundary condition of the fourth kind. The effect of input parameters in swarm algorithms on computer simulations of heat conduction was analyzed to determine the value of the coefficient κ controlling the boundary condition of the fourth kind. Two nature-inspired algorithms – bee (ABC) and ant (ACO) – were used for research purposes.

The geometry models and finite element meshes were created in GMSH software. Numerical calculations were performed using the TalyFEM [18, 19] library and algorithms implemented in C++. TalyFEM is a tool that uses the finite element method to simulate selected physical phenomena containing many data structures from the PETSc library, including vectors, matrices, or ready-made solvers [18, 19]. The tests were performed on a computer with the following parameters: an Intel(R) Core(TM) i5-4590 CPU @ 3.30 GHz, x86_64 architecture, using the Linux operating system in the Ubuntu distribution. Swarm algorithms were implemented in Python and adapted to be combined with the program based on the TalyFEM library [20]. The approximate solution error (functional) was determined and minimized using ABC and ACO's swarm algorithms. The values of T_{ij} denoted reference temperatures generated at a constant reference heat transfer coefficient κ and U_{ij} denoted the temperatures obtained during the simulation.

The research was conducted with two different finite element mesh densities, each for the same geometry model. Simulations were performed for one parameter κ, which means selecting a heat transfer coefficient from one range of values (900-1500 $\mathrm{W/m^2K}$). Reference temperatures T_{ij} were obtained for the coefficient $\kappa = 1000\ \mathrm{W/m^2K}$.

All simulations were performed for the alloy Al-2%Cu. The material properties are shown in Table 1. The initial temperatures were respectively $T_0 = 960$ K for the casting and $T_0 = 590$ K for the casting mould.
The presented results refer to the separation layer between the casting and the casting mold for different densities of finite element meshes (Fig. 1). The nodes at the interface between the casting and the casting mold have the exact spatial coordinates, which facilitated the implementation of the IV-type boundary condition in the heat conduction model.

Table 1. Material properties

	Cast	Mold
ρ [kg/m^3]	2824	7500
c [J/kgK]	1077	620
λ [W/mK]	262	40

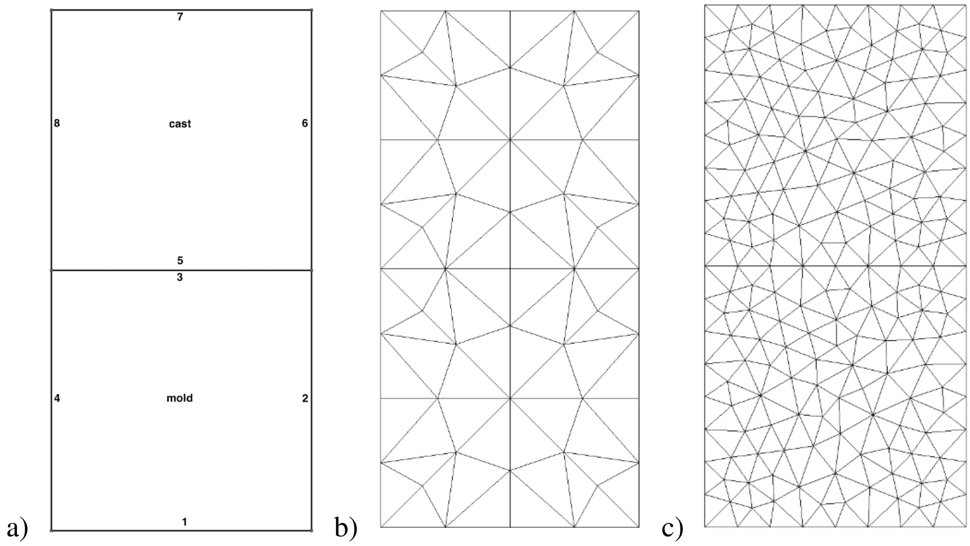

Fig. 1. Considered geometry (a) and finite element mesh with 58 nodes (b) and with 214 nodes (c)

4. Results

For each finite element grid, calculations were carried out for the two algorithms ABC and ACO, respectively, for populations equal to 5, 10, and 15 individuals and for 4 and 6 iterations. A characteristic feature of heuristic algorithms is that they must be run more than once to allow the narrowing of the search area and obtain meaningful computational results. In the article, the algorithms were run three times in each case. The 0, 1, 2, and 5% disturbance of reference values T_{ij} were also included in each case.

Table 2 presented calculations for the ABC and ACO algorithm for five individuals and a grid of 58 and 214 nodes, respectively.
The presented results show that for the tessellation of 58 nodes, better results are obtained with the ACO algorithm than with the ABC algorithm. The ACO algorithm obtained better results for six iterations with a 0 and 1% perturbation of the reference value of the κ parameter. On the other hand, after increasing the perturbation to 2 and 5%, the ACO algorithm gave better results for a smaller number of iterations.
For a finite element mesh divided into 214 nodes, generally better results are obtained with the ACO algorithm than with the ABC algorithm. Only for four iterations with 0% disturbance did the ABC algorithm obtain slightly ($\sim$0.1) better values. On the other hand, for 1 and 2% noise, both algorithms select the value of the parameter

Table 2. Values of reconstructed coefficient κ for five individuals using ABC and ACO algorithms

Noise	Iterations	κ (58 nodes)		κ (214 nodes)	
		ABC	ACO	ABC	ACO
0%	4	1007.672	1003.916	998.471	1002.382
	6	989.670	999.289	1003.503	996.786
1%	4	1094.627	1033.638	1016.864	994.710
	6	1015.832	1015.431	993.935	999.648
2%	4	1017.098	1001.587	1021.929	1021.929
	6	1014.212	1004.120	998.408	1000.635
5%	4	984.643	997.960	994.179	1011.877
	6	981.391	997.207	1009.597	1004.469

κ better for six iterations. With 5% perturbation, better results are obtained for four iterations in the case of the ABC algorithm while ACO algorithm's results are better for six iterations.

For a population of 5 individuals, the ACO's best-obtained values of the κ coefficient do not differ from the reference value by more than 1.5%.

Table 3 presented calculations for the ABC and ACO algorithm for ten individuals and a grid of 58 and 214 nodes, respectively.

Table 3. Values of reconstructed coefficient κ for ten individuals using ABC and ACO algorithms

Noise	Iterations	κ (58 nodes)		κ (214 nodes)	
		ABC	ACO	ABC	ACO
0%	4	994.009	999.667	1000.672	997.854
	6	998.689	1000.064	1012.991	999.806
1%	4	1022.261	1004.166	1008.455	999.952
	6	1005.884	1006.869	995.452	996.569
2%	4	1010.312	1006.198	997.072	995.970
	6	1015.095	1006.273	996.982	997.946
5%	4	986.919	997.315	1006.995	1004.755
	6	982.833	999.695	1019.623	1005.008

For a finite element mesh divided into 58 nodes, the ACO algorithm gives better results than the ABC algorithm. The best results are six iterations for 0 and 5% disturbance of reference values. In contrast, for 1 and 2% disturbance, the best results are for four iterations.

For a finite element mesh divided into 214 nodes, the ACO algorithm proved superior in each case. For 0 and 2% disturbance, the best results were obtained for six iterations. In contrast, for 1 and 5% disturbances, only four iterations were needed to obtain the best results.

For a population equal to 10 individuals, the ACO's best-obtained values of the κ-coefficient do not differ from the reference value by more than 0.7%.

Table 4 presented calculations for the ABC and ACO algorithm for 15 individuals and a grid of 58 and 214 nodes, respectively.

The presented results show that for the tessellation of 58 nodes, the best results were obtained after six iterations. The ACO algorithm gave better results for 0 and

Table 4. Values of reconstructed coefficient κ for 15 individuals using ABC and ACO algorithms

Noise	Iterations	κ (58 nodes)		κ (214 nodes)	
		ABC	ACO	ABC	ACO
0%	4	1001.121	999.722	1002.407	1002.347
	6	1001.020	999.826	1000.663	999.779
1%	4	1006.983	1007.422	1003.006	996.522
	6	1003.526	1006.332	1002.523	997.667
2%	4	1008.994	1006.571	997.177	998.137
	6	1005.470	1006.543	998.252	997.847
5%	4	988.039	988.039	1006.076	1004.856
	6	982.115	999.785	1004.163	1005.289

5% disturbances, while the ABC algorithm gave better results for 1 and 2% disturbances. However, the differences in the obtained κ values between ABC and ACO were not significant enough that with equal success, both algorithms can be used for calculations.

For a finite element mesh divided into 214 nodes for 0 and 1% disturbances, the best results were obtained after six iterations for the ACO algorithm. Whereas, for 2 and 5%, disturbances were also obtained after six iterations however for the ABC algorithm.

For a population equal to 15 individuals, the best κ values obtained by ABC and ACO do not differ from the reference value by more than 0.6%.

5. Conclusions

The paper analyzed the effect of input parameters on the reconstruction of the value of heat conduction coefficient κ at the interface between the casting and the casting mold. A numerical experiment was performed using bee colony (ABC) and ant colony (ACO) optimization algorithms. The input parameters were different finite element mesh densities, the number of individuals in the population, the number of iterations, and different percentage disturbances of temperature reference values.

The research results showed that all input parameters affect the value of the reconstructed heat conduction coefficient. However, increasing the number of individuals in both algorithms gives relatively lower solution errors (for 15 individuals, the errors did not exceed approximately 0.6%). The ACO algorithm gives relatively lower solution errors for a smaller number of individuals in the population. The authors showed similar trends during preliminary research presented in the article [21].

It can also be successfully stated that six iterations in the vast majority of computational cases yield a better result in the reconstruction of the coefficient κ relative to the reference values. The error of the reconstructed coefficient values almost never exceeds the magnitude of the size of introduced disturbance.

The research conducted in the article showed that both the ABC and ACO algorithms are promising tools that can find practical applications in the process of reproducing casting conditions in computer simulations.

References

[1] Rutkowski, L. (2009). *Metody i techniki sztucznej inteligencji*. Seria Informatyka – Zastosowania, Warszawa: Wydawnictwo Naukowe PWN.

[2] Słota, D. (2011). *Rozwiazywanie odwrotnych zagadnien krzepniecia z wykorzystaniem algorytmów genetycznych*. Gliwice: Wydawnictwo Politechniki Śląskiej.

[3] Changwei, M., Guangzhu, Ch., Chengliang, Y., & Yuanyuan, W. (2021). *Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm*, (vol. 156), *Computers and Industrial Engineering*, doi:https://doi.org/10.1016/j.cie.2021.107230.

[4] Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga N. (2014). A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, *42*, 21-57, https://doi.org/10.1007/s10462-012-9328-0.

[5] Komar, D. (2013). Nowa implementacja algorytmu mrówkowego wykorzystujaca technologie przetwarzania wieloprocesorowego i rozproszonego w systemie nawigacji. *Biuletyn Naukowy Wroclawskiej Wyzszej Szkoly Informatyki Stosowanej. Informatyka*, (vol. 3), 17-22, YADDA:bwmeta1.element.baztech-36e2fd37-410b-4109-8dec-b755222a1b89.

[6] Sharma, A.S., & Kim D.S. (2021). Energy efficient multipath ant colony based routing algorithm for mobile ad hoc networks. *Ad Hoc Networks*, 113, doi:https://doi.org/10.1016/j.adhoc.2020.102396.

[7] Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report*. ID: 8215393.

[8] Gerardo, B., & Wang, J. (1989). Swarm Intelligence. *Proceedings of the Seventh Annual Meeting of the Robotic Society of Japan*. RSJ Press, 425-428.

[9] Hackwood S., & Beni G. (1992). *Self-organization of sensors for swarm intelligence*. Proceedings of IEEE International Conference on Robotics and Automation, 819-829.

[10] Hetmaniok, E., Słota, D., & Zielonka, A. (2012). *Application of the Ant Colony Optimization Algorithm for Reconstruction of the Thermal Conductivity Coefficient*. Swarm and Evolutionary Computation, Proceedings. Zakopane, I, 240-244.

[11] Hetmaniok, E., Słota, D., & Zielonka, A. (2013). *Application of the Swarm Intelligence Algorithm for Investigating the Inverse Continuous Casting Problem*. Contemporary Challenges and Solutions in Applied Artificial Intelligence, Springer International Publishing, 157-162.

[12] Gorecki, J. (2021). *Preliminary analysis of the sensitivity of the FEM model of the process of dry ice extrusion in the die with a circularly converging channel on the changing its geometrical parameters*. IOP Conference Series: Materials Science and Engineering, IOP Conf. Series: Materials Science and Engineering, 1199, 012006, DOI: 10.1088/1757-899X/1199/1/012006.

[13] Berdychowski, M., Gorecki, J., Biszczanik, A., & Wałesa, K. (2022). Numerical simulation of dry ice compaction process: Comparison of Drucker-Prager/cap and cam clay models with experimental results. *Materials*, *15*, DOI: 10.3390/ma15165771.

[14] Zheng, Q., Xiao, Y., Zhang, T., Zhu, P., Ma, W., & Liu, J. 2020. Numerical simulation of latent heat of solidification for low pressure casting of aluminum alloy wheels. *Metals*, *10*, doi:10.3390/met10081024.

[15] Wisniewski, S., & Wisniewski. T.S. (2012). *Wymiana ciepła*. Warszawa: WNT.

[16] Sczygiol, N. (2000). *Modelowanie numeryczne zjawisk termomechanicznych w krzepnacym odlewie i formie odlewniczej*. Częstochowa: Wydawnictwo Politechniki Czestochowskiej.

[17] Karaboga, D. & Basturk, B. (2007). *Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems*. Berlin Heidelberg: Springer-Verlag, 789-798.

[18] Dyja, R. & Grosser, A. (2012). Obliczanie równoległe w symulacji krzepnięcia wykorzystującej model pośredni narastania fazy stałej. *Applications of Physics in Mechanical and Material Engineering*, *42*, 1896-771Z, DOI: 10.1016/j.xxx.2012.08.007.

[19] Dyja, R., Gawronska, E., Grosser, A., Jeruszka, P., & Sczygiol, N. (2016). Estimate the impact of different heat capacity approximation methods on the numerical results during computer simulation of solidification. *Engineering Letters*.

[20] Dyja, R. (2021). Comparison of results from in-house solidification convection model with standard benchmark. *Applications of Physics in Mechanical and Material Engineering*, DOI: 10.12693/APhysPolA.XX.TEMP-9502.

[21] Gawronska, G., Dyja, R., Zych, M., & Domek, G. (2022). Selection of the heat transfer coefficient using swarming algorithms. *Acta Mechanica et Automatica*, *16*, 4, Special Issue "Machine Modeling and Simulations 2022", DOI: 10.2478/ama-2022-0039.