

Article

# Comparison of the Probabilistic Ant Colony Optimization Algorithm and Some Iteration Method in Application for Solving the Inverse Problem on Model With the Caputo Type Fractional Derivative

Rafał Brociek <sup>1,†,\*</sup> , Agata Chmielowska <sup>1,†</sup>  and Damian Słota <sup>1,†</sup> 

Department of Mathematics Applications and Methods for Artificial Intelligence, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland; agata.chmielowska@polsl.pl (A.C.); damian.slota@polsl.pl (D.S.)

\* Correspondence: rafal.brociek@polsl.pl

† These authors contributed equally to this work.

Received: date; Accepted: date; Published: date

**Abstract:** This paper presents the algorithms for solving the inverse problems on models with the fractional derivative. The presented algorithm is based on the Real Ant Colony Optimization algorithm. In this paper, the examples of the algorithm application for the inverse heat conduction problem on the model with the fractional derivative of the Caputo type is also presented. Based on those examples, the authors are comparing the proposed algorithm with the iteration method presented in the paper: Zhang, Z. An undetermined coefficient problem for a fractional diffusion equation. *Inverse Probl.* 2016, 32.

**Keywords:** fractional derivative; fractional differential equation; inverse problem; mathematical modeling; heat conduction

## 1. Introduction

The inverse problems are an important and commonly encountered class of problems in many branches of technology and mathematics [1,2]. Solving them involves in the appropriate choice of the input parameters of the model (e.g., material coefficients or initial-boundary conditions) in order to model the process in such a way that the suitable output is obtained.

In this paper, we consider the model with the fractional differential equation with the fractional derivative [3,4] of the Caputo type. This type of derivative is being used to describe the phenomenon of the anomalous diffusion. The example of this kind of phenomena is a heat conduction in the porous material. Voller [5] shows that for the materials of anomalous porosity, the models containing the fractional derivative are better for describing the problem than the models with the classical derivative. The experiments conducted in the papers [6,7] show that the fractional derivatives make sense when describing the phenomenon of the heat conduction in such materials as the composites or the porous aluminum. In [6], the authors consider the inverse heat conduction problem in the composite material. For the model presented in this paper the fractional derivative of the Caputo type was used. In [7], the authors are comparing the mathematical models containing various derivatives (classical, fractional of the Riemann–Liouville type and fractional of the Caputo type) basing on the experimental data obtained for the porous aluminum. Moreover, in this case the fractional derivative turned out to be better for describing the phenomenon. More on the application of the fractional calculus in various fields of science can be found in the papers [8–14].

In this paper, we present the algorithm for solving the inverse problem on the model containing the fractional derivative of the Caputo type. In the presented model, the thermal conductivity coefficient is

being reconstructed. For solving the direct problem the finite difference method with the approximation of the Caputo type derivative [15] was used. The algorithm for solving the inverse problem consists in minimizing the functional representing the approximation error. For this purpose, one of the algorithms of the swarm intelligence was used [16,17]. In this paper, the numerical examples used for comparison of the presented algorithm with the algorithm shown in [18] are also being presented.

## 2. Formulation of the Problem

We consider two methods for solving the inverse problem on the models containing the fractional derivatives. Let us consider the following differential equation with the fractional derivative and the initial-boundary conditions.

$$c\rho \frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \lambda(t) \frac{\partial^2 u(x,t)}{\partial x^2} + g(x,t), \quad (1)$$

$$u(x,0) = f(x), \quad x \in [0, L], \quad (2)$$

$$u(0,t) = \phi(t), \quad t \in (0, T), \quad (3)$$

$$u(L,t) = \psi(t), \quad t \in (0, T). \quad (4)$$

In the above Equation there is a fractional derivative with respect to time. In this paper, this derivative is of order  $\alpha \in (0,1)$  and of a Caputo type which is defined as follows,

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x,s)}{\partial s} (t-s)^{-\alpha} ds. \quad (5)$$

Equation (1) (with the fractional derivative of order  $\alpha$  with respect to time) with the initial-boundary conditions (2)–(4) may be used to model the heat conduction phenomenon in the porous material. Then, the respective components of the model take the following names.

$x, t$ —spatial and time variable,

$u$ —function representing the temperature distribution,

$c$ —specific heat,

$\rho$ —density,

$\lambda$ —thermal conductivity,

$\alpha$ —order of fractional derivative,

$g$ —additional heat source,

$f$ —function representing the initial condition,

$\phi, \psi$ —functions representing the boundary condition of the first type.

Generally speaking, the solution of the inverse heat conduction problem consists in such a choice of the model parameters (model input) so the given output temperature distribution is obtained. More information regarding the heat processes modeling in the porous materials, inverse heat conduction problems, and use of the fractional calculus for modeling these kind of processes may be found in [5,7,19–21].

### 2.1. Description of the Considered Inverse Problem

In this paper, we consider two algorithms for solving the inverse problem. The description of the first method can be found in the paper [18]. This method is iterative so in this paper we will call it the iteration method. The second algorithm is the algorithm proposed by the authors of this paper and it is based on the Real Ant Colony Optimization algorithm [16], which is the swarm artificial intelligence algorithm. The details of the algorithm are being described in the Section 3.2. More about application of the swarm intelligence algorithms can be found in [17,22,23].

In the example the thermal conductivity dependent on time  $\lambda$  coefficient which occurs in the Equation (1) is being reconstructed. The additional information on the basis of which the coefficient  $\lambda$  is being searched is the condition

$$-\lambda(t) \left( \frac{\partial u}{\partial x}(0, t) - a(t) \right) = b(t), \quad t \in (0, T],$$

where the function  $a$  is given and the function  $b$  (the values of the function  $b$  at the points of the grid) is being calculated from the formula

$$b(t^k) = b^k = -\lambda^k \left( \frac{u_1^k - u_0^k}{\Delta x} - a^k \right), \tag{6}$$

by first solving the direct problem for the exact value of the parameter  $\lambda$ . The values  $b^k$  are treated as the input data for the inverse problem. For the determined thermal conductivity coefficient  $\bar{\lambda}$  by solving the direct problem we obtain the values  $u_0^k, u_1^k$ , and therefore we also obtain the values  $\bar{b}^k$ . By comparing the values  $b^k$  and  $\bar{b}^k$  we construct the algorithm:

$$J(\bar{\lambda}) = \sum_{k=1}^{\bar{K}} \left( \bar{b}^k - b^k \right)^2. \tag{7}$$

By minimizing this functional (using the algorithm proposed by the authors) we are reconstructing the thermal conductivity coefficient  $\lambda$ .

### 3. Methods of Solution

In the process of solving the described inverse problem there is a need to repeatedly solve the direct problem. In this section, we first describe the solution of the direct problem and then describe the Real Ant Colony Optimization algorithm needed for minimizing the functional (7).

#### 3.1. Solution of the Direct Problem

The direct problem described by the Equation (1) and the initial-boundary conditions (2)–(4) were resolved by the use of the finite difference method.

Let  $N, K \in \mathbb{N}$  be the size of the grid respectively in the domain of space and time. We take the following grid steps;  $\Delta x = L/N, \Delta t = T/K$ . Then, the points of the grid from the space interval  $[0, L]$  are the numbers  $x_i = i\Delta x, i = 0, 1, 2, \dots, N$  and from the time interval  $[0, T]$  the numbers  $t_k = k\Delta t, k = 0, 1, 2, \dots, K$ . The values of the functions  $f, g, u, \phi, \psi$  at the grid points are denoted as follows:  $f_i = f(x_i), g_i^k = g(x_i, t_k), \phi^k = \phi(t_k), \psi^k = \psi(t_k)$  and  $u_i^k = u(x_i, t_k)$ . The values of the approximation function at the points  $(x_i, t_k)$  are denoted as  $U_i^k$ . The Equation (1) may be written as follows,

$$\frac{\partial^\alpha u(x, t)}{\partial t^\alpha} = a \frac{\partial^2 u(x, t)}{\partial x^2} + \bar{g}(x, t), \tag{8}$$

where  $a = \frac{\lambda}{c\rho}$  in terms of the heat conductivity is the thermal diffusivity coefficient and  $\bar{g}(x, t) = \frac{g(x, t)}{c\rho}$ . The fractional derivative of the Caputo type of the order  $\alpha \in (0, 1)$  with respect to time is approximated by the following Equation [15],

$$\frac{\partial^\alpha u(x, t)}{\partial t^\alpha} \approx \sigma(\alpha, \Delta t) \sum_{j=1}^k \omega(\alpha, j) (U_i^{k-j+1} - U_i^{k-j}), \tag{9}$$

where

$$\sigma(\alpha, \Delta t) = \frac{1}{\Gamma(1 - \alpha)(1 - \alpha)\Delta t^\alpha},$$

$$\omega(\alpha, j) = j^{1-\alpha} - (j - 1)^{1-\alpha}.$$

Using the Equation (9), the boundary conditions  $U_0^k = \phi^k$ ,  $U_N^k = \psi^k$ ,  $k \geq 1$  and the differential quotient for the second derivative  $\frac{\partial^2 u(x_i, t_k)}{\partial x^2} \approx \frac{1}{(\Delta x)^2} (U_{i-1}^k - 2U_i^k + U_{i+1}^k)$ , after proper organizing the elements we obtain the following discrete differential equations ( $k \geq 1, i = 1, 2, \dots, N - 1$ ):

$$\begin{aligned} & -\frac{a}{(\Delta x)^2} U_{i-1}^k + \left( \sigma(\alpha, \Delta t) + \frac{2a}{(\Delta x)^2} \right) U_i^k - \frac{a}{(\Delta x)^2} U_{i+1}^k = \\ & = \sigma(\alpha, \Delta t) U_i^{k-1} - \sigma(\alpha, \Delta t) \sum_{j=2}^k \omega(\alpha, j) (U_i^{k-j+1} - U_i^{k-j}) + \bar{g}_i^k. \end{aligned} \tag{10}$$

The Equation (10) may be also written in the matrix form

$$AU^k = D, \quad k = 1, 2, \dots, K \tag{11}$$

where  $U^k = [U_1^k, U_2^k, \dots, U_{N-1}^k]^T$ ,  $A$  is a tridiagonal matrix of the size  $(N - 1) \times (N - 1)$  and following coefficients,

$$i = 1, 2, \dots, N - 1, \quad j = 1, 2, \dots, N - 1,$$

$$a_{ij} = \begin{cases} \sigma(\alpha, \Delta t) + \frac{2a}{(\Delta x)^2}, & j = i, \\ -\frac{2a}{(\Delta x)^2}, & (j = i + 1 \wedge i = 1, 2, \dots, N - 2) \vee \\ & (j = i - 1 \wedge i = 2, 3, \dots, N - 1), \end{cases}$$

and  $D$  is a vector  $[d_1 + \frac{2a}{(\Delta x)^2} \phi^k, d_2, d_3, \dots, d_{N-2}, d_{N-1} + \frac{2a}{(\Delta x)^2} \psi^k]^T$ , where:

$$d_i = \sigma(\alpha, \Delta t) U_i^{k-1} - \sigma(\alpha, \Delta t) \sum_{j=2}^k \omega(\alpha, j) (U_i^{k-j+1} - U_i^{k-j}) + \bar{g}_i^k.$$

### 3.2. Solution of the Inverse Problem

In order to minimize the functional (7), the Real Ant Colony Optimization algorithm [16] is being used. The inspiration for the algorithm was the swarming behavior of ants in nature. In the presented algorithm, the role of the solutions are being played by the pheromone spots in the number of  $L$ . At the beginning those spots are randomly placed in the considered region. Next, they are being sorted by the quality and certain probabilities are being assigned to them. The better the solution is, the higher the probability of choosing it is. This way we create the set of solutions. In every iteration each of  $M$  ants construct one new solution (new pheromone spot) using the probability density function (Gauss function). First the ant chooses with the certain probability the solution that will be transformed. This probability is dependent on the parameter  $q$  of the algorithm. If value of  $q$  is small then the choice of the best solution is preferred and the higher the value of  $q$  is, the more similar to each other the probabilities of choosing any solution are. When the solution is chosen then the ant samples its neighborhood using the Gauss function with the parameters  $\mu$  and  $\sigma$ , where  $\sigma$  is dependent on the parameter  $\zeta > 0$ . This parameter is responsible for the disappearance of the pheromone trace. The smaller value of  $\zeta$  is, the faster the algorithm converges to its best solution at the expense of further exploration of the region. Then, the set of solutions is updated with the new solutions—the new and previous solutions are being sorted by quality and the  $M$  of the worst solutions get rejected. Moreover, the algorithm was adapted for parallel computing. Mathematical fundamentals and results concerning the convergence of the different versions of the ACO algorithm are included in [24–27]. Based on the analysis of the papers of various authors, see for example [28–32], as well as on the computation carried out by us (for example [7,17,20,22,31]) we come to the conclusion that using the

artificial intelligence algorithms (ACO, RACO, RACO, IRM, etc.) in most cases does not require the application of the explicit regulatory procedures. There is no mathematical proof for that, only the empirical data. For the examples presented in the paper we have known the exact solutions so we could compare the approximate solutions with them. The reconstruction errors of the reconstructed parameters generally do not exceed the input data errors. They are slightly larger in some cases. It follows that the results obtained for the disturbed input data remain stable. In order to describe the algorithm we are introducing the notation

$F$  – minimized function (the objective function),

$n$  – number of objective function variables (number of searched parameters),

$nT$  – number of threads,  $M = nT \times p$  – number of ants in the population,

$I$  – number of iterations,  $L$  – number of pheromone spots,  $q, \xi$  – algorithm parameters.

We will present the consecutive steps of the algorithm.

### Initialization of the algorithm

1. Setting the input parameters of the algorithm:  $L, M, I, nT, q, \xi$ .
2. Generating  $L$  solutions playing the role of the pheromone spots. Assigning them to the set  $T_0$  (initial archive).
3. Computing the values of the objective function for all of the pheromone spots (parallel computing) and sorting the archive  $T_0$  from the best solution to the worst.

### Iteration process

4. Assigning the probabilities to the pheromone spots according to the pattern:

$$p_l = \frac{\omega_l}{\sum_{l=1}^L \omega_l} \quad l = 1, 2, \dots, L,$$

where weights  $\omega_l$  are related to the solution number  $l$  and are expressed by the formula:

$$\omega_l = \frac{1}{qL\sqrt{2\pi}} \times e^{-\frac{(l-1)^2}{2q^2L^2}}.$$

5. The ant randomly chooses the  $l$ -th solution with the probability  $p_l$ .
6. The ant transforms the  $j$ -th coordinate ( $j = 1, 2, \dots, n$ ) of the  $l$ -th solution  $s_j^l$  by sampling the neighborhood using the probability density function (in this case the Gauss function):

$$g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where  $\mu = s_j^l$ ,  $\sigma = \frac{\xi}{L-1} \sum_{p=1}^L |s_j^p - s_j^l|$ .

7. Steps 5–6 are being repeated by every ant. We get  $M$  new solutions (pheromone spots) by that.
8. Partition of the new solutions into the  $nT$  groups. Computing the value of the objective function for the new solutions (parallel computing).
9. Adding the new solutions to the archive  $T_i$ , sorting them by the quality and rejecting the  $M$  worst solutions.
10. The steps 3–9 are being repeated  $I$  times.

Knowing the values of the parameters  $L, M$  and  $I$  we can determine the number of the objective function calls during the algorithm execution. This number is  $L + M \cdot I$ . The Figure 1 presents the block diagram of the RealACO algorithm.

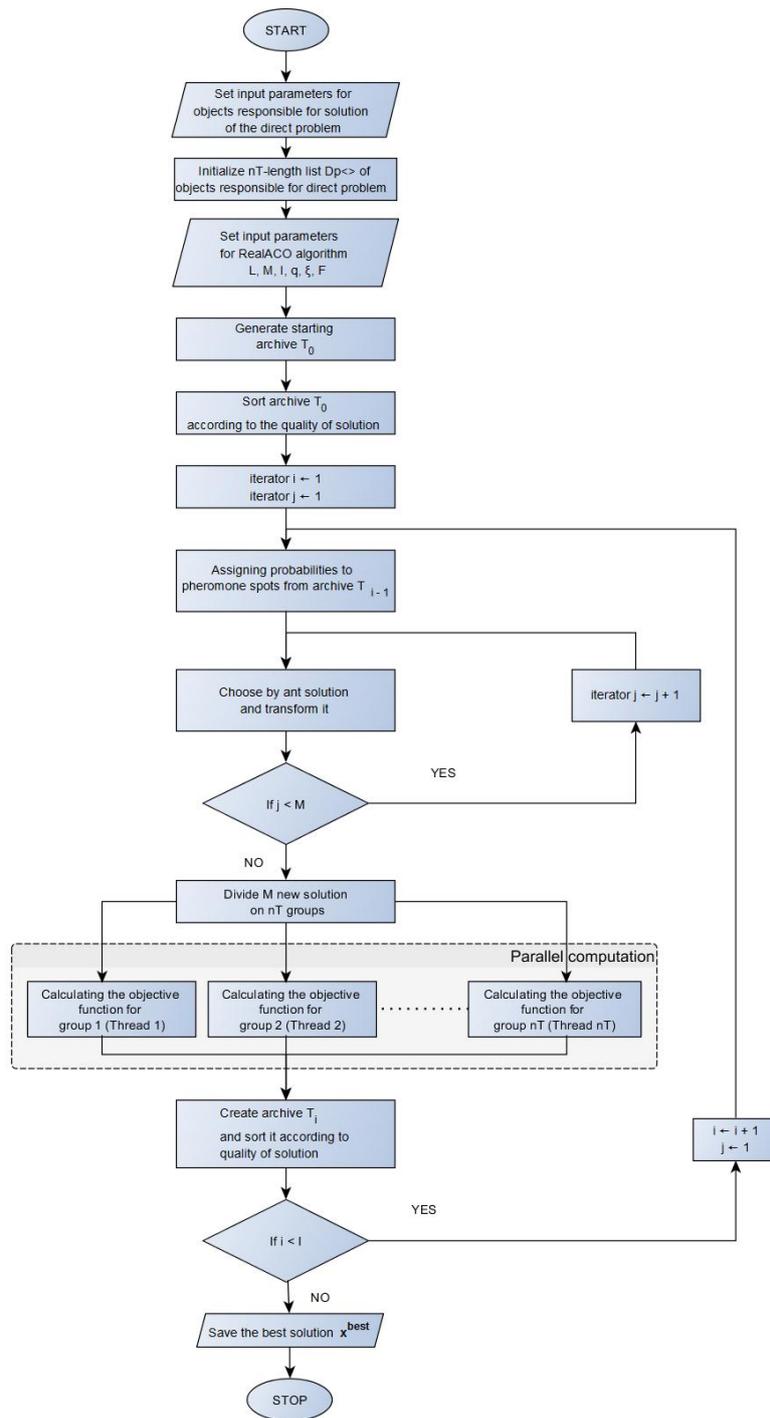


Figure 1. The block diagram of the Real Ant Colony Algorithm.

#### 4. Numerical Examples

In this section, we present two numerical examples that we used for comparing the algorithm proposed by the authors (based on the Real Ant Colony Optimization algorithm) with the iteration algorithm presented in the paper [18]. In both examples we are considering the model (1)–(4), and the reconstructed coefficient is thermal conductivity  $\lambda$  dependent on time. The input data for the inverse problem is a set of values  $b^k$  (6).

##### 4.1. Example 1

For this example we take the following data:

$$f(x) = 0, \quad \phi(t) = \psi(t) = 0, \quad \alpha = 0.9, \quad c = \varrho = 1,$$

$$L = T = 1, \quad a(t) = t^2 + t, \quad g(x, t) = 0.105114(10t^{0.1} + 18.1818t^{1.1})(x - 1).$$

The reconstructed coefficient  $\lambda$  is in the following form:

$$\lambda(t) = \begin{cases} a_1 & t \in (0, \frac{1}{3}), \\ a_2 & t \in [\frac{1}{3}, \frac{2}{3}), \\ a_3 & t \in [\frac{2}{3}, 1]. \end{cases}$$

The exact values of the parameters  $a_1, a_2, a_3$  are 1, 0.5, 1. For minimizing the functional (7) the RealACO was applied with the parameters  $L = 6, M = 15, I = 40$ . The values of the parameters were chosen basing on the experiments that we had carried out earlier. The number of the objective function calls is 606. Table 1 presents the results of the reconstruction.

**Table 1.** Results of calculations for the RealACO algorithm (grid  $100 \times 900$ ) ( $\bar{a}_i$ —reconstructed value of the coefficient  $a_i$ ,  $\delta_{\bar{a}_i}$ —the relative error of reconstruction of the coefficient  $a_i$ ,  $\sigma$ —standard deviation ( $i = 1, 2, 3$ )).

Noise	$\bar{a}_i$	$\delta_{\bar{a}_i}$ [%]	$\sigma$
0%	0.9999	0.01	$4.36 \cdot 10^{-4}$
	0.4999	0.02	$7.81 \cdot 10^{-5}$
	0.9999	0.01	$4.36 \cdot 10^{-5}$
1%	1.0001	0.01	$1.77 \cdot 10^{-4}$
	0.4998	0.04	$1.74 \cdot 10^{-4}$
	0.9999	0.01	$7.21 \cdot 10^{-5}$
2%	0.9995	0.05	$1.96 \cdot 10^{-4}$
	0.4996	0.08	$3.67 \cdot 10^{-4}$
	0.9997	0.03	$1.96 \cdot 10^{-5}$
5%	1.0001	0.01	$3.79 \cdot 10^{-4}$
	0.4996	0.08	$5.82 \cdot 10^{-5}$
	1.0003	0.03	$4.84 \cdot 10^{-5}$

The error of the obtained results was calculated according to formula

$$\text{error} = \sqrt{\sum_{k=1}^N |\bar{\lambda}^k - \lambda^k|^2},$$

where  $N$  indicates the number of measurements,  $\bar{\lambda}$  is the reconstructed coefficient and  $\lambda$  is the exact value of the coefficient. The grid used to generate the input data was of the size  $100 \times 900$  as well as the

grid used in the algorithm for solving the direct problem. Table 2 presents the results in dependence on the size of input data disturbance.

**Table 2.** Comparison of the algorithms according to the obtained errors of reconstructing the coefficient  $\lambda$  for the input data from the example 1.

Noise	RealACO	Iteration Method	
	Error	$\epsilon$	Error
0%	$4.2781 \cdot 10^{-4}$	0.01	$4.8035 \cdot 10^{-3}$
		0.001	$4.0918 \cdot 10^{-4}$
		0.0001	$3.5545 \cdot 10^{-5}$
1%	$3.3106 \cdot 10^{-3}$	0.01	$9.0230 \cdot 10^{-2}$
		0.001	$8.9619 \cdot 10^{-2}$
		0.0001	$8.9578 \cdot 10^{-2}$
2%	$1.0889 \cdot 10^{-2}$	0.01	$1.8033 \cdot 10^{-1}$
		0.001	$1.8057 \cdot 10^{-1}$
		0.0001	$1.8059 \cdot 10^{-1}$
5%	$8.7965 \cdot 10^{-3}$	0.01	$4.4052 \cdot 10^{-1}$
		0.001	$4.4016 \cdot 10^{-1}$
		0.0001	$4.4012 \cdot 10^{-1}$

In case of the exact input data the smaller value of the parameter  $\epsilon$  is, the smaller error of the result obtained by the iteration algorithm we get. For  $\epsilon = 0.001$ , the approximate error is  $4.1 \cdot 10^{-4}$ , while for  $\epsilon = 0.0001$ , the error is  $3.56 \cdot 10^{-5}$ . In case of the RealACO algorithm the obtained error is approximately  $4.28 \cdot 10^{-4}$  and it is slightly bigger than the error obtained by the iteration algorithm. For the input data disturbed by the pseudorandom error decreasing the parameter  $\epsilon$  in the iteration method slightly reduces the errors of the obtained results. In case of the disturbed input data the reconstruction errors of the thermal conductivity coefficient  $\lambda$  when using the algorithm proposed in the paper were smaller than when using the iteration method. For the data disturbed by the 5% pseudo-random error the RealACO algorithm reconstructed the  $\lambda$  coefficient with the error  $8.7965 \cdot 10^{-3}$  while for the iteration method the error was  $4.4012 \cdot 10^{-1}$ .

#### 4.2. Example 2

The input data used in the previous example were taken from the paper [18]. On the contrary, this example was not presented in that paper. In this example we compare both algorithms with the input data generated on a grid of a different density than the grid used in the algorithms reconstructing the  $\lambda$  coefficient. In such a case, testing the algorithm is a base for avoiding the inverse crimes [33] (page 5). The grid used for generating the input data was of size  $100 \times 2000$  and the grid used in the reconstruction process was of size  $100 \times 1800$ . The algorithm described in the paper [18] was implemented by us in the C# and then the computation was carried out. The results are presented in the Table 3.

The computational data are as follows.

$$f(x) = 0, \quad \phi(t) = \psi(t) = 0, \quad \alpha = 0.9, \quad c = \rho = 1,$$

$$L = T = 1, \quad a(t) = t^2 + t, \quad g(x, t) = 0.105114(10t^{0.1} + 18.1818t^{1.1})(x - 1).$$

The  $\lambda$  coefficient is being reconstructed in the following form,

$$\lambda(t) = a_1 \sin(a_2 \pi t) + a_3,$$

where the exact values of the parameters  $a_1$ ,  $a_2$ , and  $a_3$  are 1, 4, and 1.1, respectively. In the RealACO algorithm, the parameters  $M = 16$ ,  $L = 12$ , and  $I = 35$  were used. The values of the parameters were chosen basing on the experiments that we had carried out earlier. Table 4 presents the results obtained for the RealACO algorithm.

**Table 3.** Comparison of the algorithms according to the obtained errors of reconstructing the coefficient  $\lambda$  for the input data from example 2.

Noise	RealACO	Iteration Method	
	Error	$\epsilon$	Error
0%	0.6230	0.01	1.1821
		0.001	1.1825
		0.0001	1.1826
1%	0.5982	0.01	1.2050
		0.001	1.2054
		0.0001	1.2055
2%	0.6146	0.01	1.2369
		0.001	1.2374
		0.0001	1.2375
5%	0.6566	0.01	1.4903
		0.001	1.4907
		0.0001	1.4908

The result of the reconstruction obtained from the RealACO algorithm are very well, the most reconstruction errors are smaller than the input data errors. Even for the input data disturbed by the 5% error the reconstruction errors are no more that 1.19%. Moreover, the values of the standard deviation for the obtained results are low.

Table 3 presents the results used for comparison of the algorithms for various disturbances of the input data. In case of the iteration method the change of the  $\epsilon$  parameter insignificantly affects the reconstruction errors of the coefficient  $\lambda$ . In each considered case the reconstruction errors of the  $\lambda$  coefficient are smaller for the RealACO algorithm than for the iteration method.

**Table 4.** Results obtained by the RealACO algorithm (grid  $100 \times 1800$ ) ( $\bar{a}_i$  – reconstructed value of a coefficient  $a_i$ ,  $\delta_{\bar{a}_i}$  – relative error of reconstruction of the coefficient  $a_i$ ,  $\sigma$  – standard deviation ( $i = 1, 2, 3$ )).

Noise	$\bar{a}_i$	$\delta_{\bar{a}_i}$ [%]	$\sigma$
0%	0.9889	1.10	0.1077
	3.9999	0.01	0.0124
	1.0875	1.14	0.0507
1%	0.9910	0.90	0.0128
	3.9991	0.03	0.0040
	1.0874	1.15	0.0140
2%	0.9900	0.99	0.0266
	3.9998	0.01	0.0076
	1.0873	1.15	0.0156
5%	0.9881	1.18	0.0277
	4.0002	0.01	0.0182
	1.0869	1.18	0.0150

Finally, we present how the solutions in the respective iterations converge to the approximate solution for the Real Ant Colony Optimization algorithm (Figures 2 and 3). From about the twentieth iteration the value of the reconstructed coefficient as well as the value of the functional start changing insignificantly.

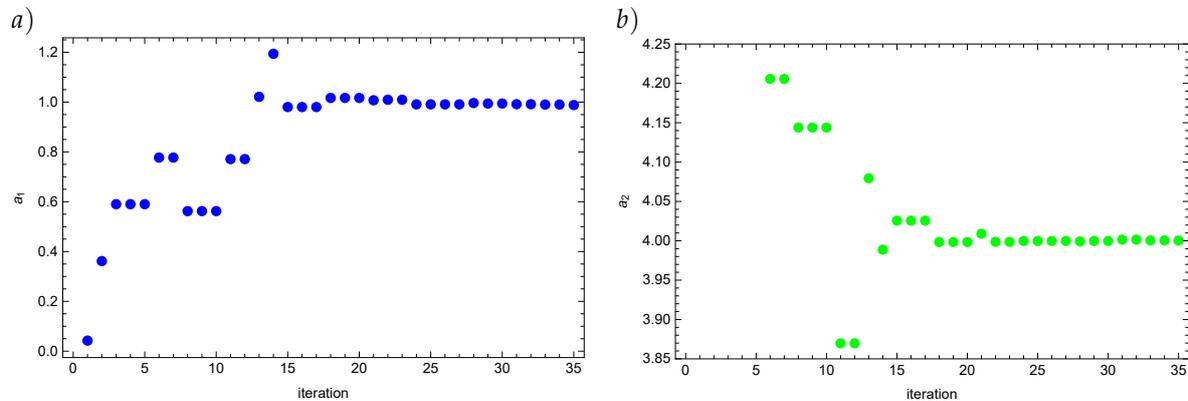


Figure 2. Values of the coefficients  $\bar{a}_1$  (a) and  $\bar{a}_2$  (b) for the input data disturbed by the 5% error.

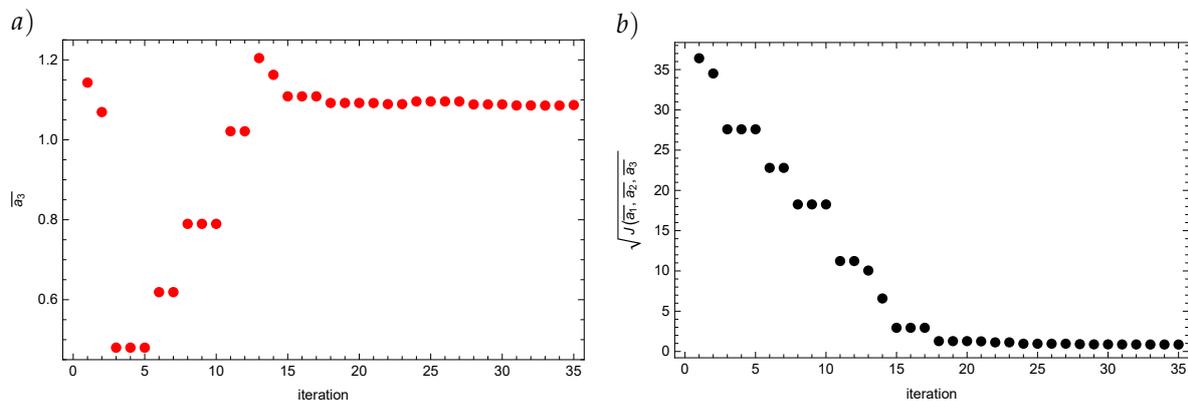


Figure 3. Value of the coefficient  $\bar{a}_3$  (a) and value of the functional  $J$  (b) for the input data disturbed by the 5% error.

### 5. Conclusions

The algorithm for solving the inverse problems on the models described by the differential equation with the Caputo type derivative presented in the paper seems to be an effective tool for solving these type of problems. The presented method comes to minimizing the functional representing the approximation error. For the functional, the classical methods of finding the extrema cannot be used and because of that the Real Ant Colony Optimization algorithm was used in the paper. The reconstruction errors obtained in the presented examples were small and did not exceed the input data disturbances. Additionally, the presented method turned out to be more precise than the method presented in the article [18], especially assuming that the input data was generated on different grid than the grid used for solving the direct problem.

**Author Contributions:** Conceptualization, D.S., R.B., and A.C.; methodology, D.S.; software, R.B.; validation, A.C.; investigation, R.B.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Miller, J.L.; Siltanen, S. *Linear and Nonlinear Inverse Problems with Practical Applications*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA 2012.
2. Vogel, C.R. *Computational Methods for Inverse Problems*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA 2002.
3. Ortigueira, M.D.; Tenreiro Machado, J.A. What is a fractional derivative? *J. Comput. Phys.* **2015**, *293*, 4–13. [[CrossRef](#)]
4. Miller, K.; Ross, B. *An Introduction to the Fractional Calculus and Fractional Differential*; Wiley: New York, NY, USA, 1993.
5. Voller, V.R. Anomalous heat transfer: Examples, fundamentals, and fractional calculus models. *Adv. Heat Transf.* **2018**, *50*, 338–380.
6. Zhuang, Q.; Yu, B.; Jiang, X. An inverse problem of parameter estimation for time-fractional heat conduction in a composite medium using carbon-carbon experimental data. *Phys. B* **2015**, *456*, 9–15. [[CrossRef](#)]
7. Brociek, R.; Słota, D.; Król, M.; Matula, G.; Kwaśny, W. Comparison of mathematical models with fractional derivative for the heat conduction inverse problem based on the measurements of temperature in porous aluminum. *Int. J. Heat Mass Transf.* **2019**, *143*, 118440. [[CrossRef](#)]
8. Yang, X.J.; Hristov, J.; Srivastava, H.M.; Ahmad, B. Modelling fractal waves on shallow water surfaces via local fractional Korteweg-de Vries equation. *Abs. Appl. Anal.* **2014**, Article ID 278672. [[CrossRef](#)]
9. Hristov, J. Approximate solutions to time-fractional models by integral-balance approach. *Fract. Dyn.* **2015**, 78–109.
10. Sowa, M. A harmonic balance methodology for circuits with fractional and nonlinear elements. *Circuits Syst. Signal Process.* **2018**, *37*, 4695–4727. [[CrossRef](#)]
11. Oprzędkiewicz, K.; Mitkowski, W. A memory-efficient noninteger-order discrete-time state-space model of a heat transfer process. *Int. J. Appl. Math. Comput. Sci.* **2018**, *28*, 649–659. [[CrossRef](#)]
12. Toufik, M., Atangana A., New numerical approximation of fractional derivative with non-local and non-singular kernel: Application to chaotic models. *Eur. Phys. J. Plus* **2017**, *132*, 444. [[CrossRef](#)]
13. Yang, X.-J.; Tenreiro Machado, J.A. A new fractional operator of variable order: Application in the description of anomalous diffusion. *Phys. A: Stat. Mech. Its Appl.* **2017**, *481*, 276–283. [[CrossRef](#)]
14. Koleva, M.N.; Vulkov, L.G. Numerical solution of time-fractional Black-Scholes equation. *Comput. Appl. Math.* **2017**, *36*, 1699–1715. [[CrossRef](#)]
15. Murio, D.A. Implicit finite difference approximation for time fractional diffusion equations. *Comput. Math. Appl.* **2008**, *56*, 1138–1145. [[CrossRef](#)]
16. Socha, K.; Dorigo, M. Ant colony optimization for continuous domains. *Eur. J. Oper. Res.* **2008**, *185*, 1155–1173. [[CrossRef](#)]
17. Hetmaniok, E.; Słota, D.; Zielonka, A. Experimental verification of selected artificial intelligence algorithms used for solving the inverse Stefan problem. *Numer. Heat Transf. Part B-Fundam.* **2014**, *66*, 343–359. [[CrossRef](#)]
18. Zhang, Z. An undetermined coefficient problem for a fractional diffusion equation. *Inverse Probl.* **2016**, *32*, 015011. [[CrossRef](#)]
19. Baleanu, D.; Yang, X.J. Fractal heat conduction problem solved by local fractional variation iteration method. *Therm. Sci.* **2013**, *17*, 625–628.
20. Brociek, R.; Słota, D.; Król, M.; Matula, G.; Kwaśny, W. Modeling of heat distribution in porous aluminum using fractional differential equation. *Fractal Fract.* **2017**, *1*, 1–9. [[CrossRef](#)]
21. Voller, V.R. Computations of anomalous phase change. *Int. J. Numer. Methods Heat Fluid Flow* **2016**, *26*, 624–638. [[CrossRef](#)]
22. Brociek, R.; Słota, D. Application of Real Ant Colony Optimization algorithm to solve space and time fractional heat conduction inverse problem. *Inf. Technol. Control* **2017**, *46*, 171–182. [[CrossRef](#)]
23. Mavrovouniotis, M.; Li, Ch.; Yang, S. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm Evol. Comput.* **2017**, *33*, 1–17. [[CrossRef](#)]
24. Gutjahr, W.J. ACO algorithms with guaranteed convergence to the optimal solution. *Inf. Process. Lett.* **2002**, *82*, 145–153. [[CrossRef](#)]
25. Dorigo, M.; Blum, Ch. Ant colony optimization theory: A survey. *Theor. Comput. Sci.* **2005**, *344*, 243–278. [[CrossRef](#)]

26. Duan, H. Ant colony optimization: Principle, convergence and application. In *Handbook of Swarm Intelligence. Adaptation, Learning, and Optimization*; Panigrahi, B.K., Shi, Y., Lim, M.-H., Eds.; Springer: Berlin, Germany, 2011; pp. 373–388.
27. Chen, W.; Bian, W.; Zeng, R. Research on convergence of ACO subset algorithms *COMPEL: Int. J. Comput. Math. Electr. Electron. Eng.* **2013**, *32*, 649–660. [[CrossRef](#)]
28. Fainekos, G.E.; Giannakoglou, K.C. Inverse design of airfoils based on a novel formulation of the ant colony optimization method. *Inverse Probl. Eng.* **2003**, *11*, 21–38. [[CrossRef](#)]
29. Carvalho, A.R.; de Campos Velho, H.F.; Stephany, S.; Souto, R.P.; Becceneri, J.C.; Sandri, S. Fuzzy ant colony optimization for estimating chlorophyll concentration profile in offshore sea water. *Inverse Probl. Sci. Eng.* **2008**, *16*, 705–715. [[CrossRef](#)]
30. Lobato, F.S.; Steffen Jr, V.; Silva Neto, A.J. Estimation of space-dependent single scattering albedo in a radiative transfer problem using differential evolution. *Inverse Probl. Sci. Eng.* **2012**, *20*, 1043–1055. [[CrossRef](#)]
31. Hetmaniok, E.; Hristov, J.; Słota, D.; Zielonka, A. Identification of the heat transfer coefficient in the two-dimensional model of binary alloy solidification. *Heat Mass Transf.* **2017**, *53*, 1657–1666. [[CrossRef](#)]
32. Tam, J.H.; Ong, Z.Ch.; Ismail, Z.; Ang, B.Ch.; Khoo, S.Y.; Li, W.L. Inverse identification of elastic properties of composite materials using hybrid GA-ACO-PSO algorithm. *Inverse Probl. Sci. Eng.* **2018**, *26*, 1432–1463. [[CrossRef](#)]
33. Kaipio, J.; Somersalo, E. *Statistical and Computational Inverse Problems*; Springer: New York, NY, USA, 2005.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).