Reaction Kinetics, Mechanisms and Catalysis (2021) 132:111–122
https://doi.org/10.1007/s11144-021-01927-8

Gravitational search algorithm for determining the optimal kinetic parameters of propane pre-reforming reaction

L. V. Enikeeva[1,2] · D. I. Potemkin[1,3,4] · S. I. Uskov[1,3] · P. V. Snytnikov[1,3] · M. R. Enikeev[2] · I. M. Gubaydullin[2,5]

Received: 16 November 2020 / Accepted: 4 January 2021 / Published online: 19 January 2021

Abstract

The article describes the gravitational search algorithm and its application to solve the inverse problem of chemical kinetics. It is shown that metaheuristic algorithms have become more and more popular recently. In this regard, it is logical to apply these algorithms to various optimization problems. The optimization problem is formulated on the example of solving the inverse problem of chemical kinetics. The process under study is the pre-reformation of propane into a methane-rich gas on a Ni catalyst, which is an industrially important chemical process. The article briefly describes the algorithm and its pseudocode, and then compares the performance of the gravitational search algorithm with other metaheuristic methods. The algorithm showed adequate results and was applied to solve a specific technological problem. Using this algorithm, the inverse problem of chemical kinetics was solved and the optimal values of the kinetic parameters of the reaction were found. It was proved that the model correctly described the available experimental data. Moreover, the sensitivity analysis of the algorithm parameters was performed.

Keywords Mathematical modeling · Inverse problem of chemical kinetics · Gravitational search algorithm · Propane pre-reforming

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11144-021-01927-8.

✉ L. V. Enikeeva
leniza.enikeeva@yandex.ru

[1] Novosibirsk State University, 2 Pirogova St., Novosibirsk, Russia 630090

[2] Ufa State Petroleum Technological University, 1 Kosmonavtov St., Ufa, Russia 450062

[3] Boreskov Institute of Catalysis SB RAS, 5 Lavrentiev Ave, Novosibirsk, Russia 630090

[4] Novosibirsk State Technical University, 20 pr. K. Marksa, Novosibirsk 630073, Russia

[5] Institute of Petrochemistry and Catalysis, Russian Academy of Sciences, 141 pr. Oktyabrya, Ufa, Russia 450075

Introduction

Optimization problems can be found in almost all engineering fields. Thus, the development of optimization techniques is very essential for engineering applications. Most traditional optimization methods require information about the gradient, and therefore they cannot be used to solve non-differentiable functions. Furthermore, such methods usually suffer from falling into the local optimum when solving complex optimization problems with many local optima. However, many real-world engineering optimization problems are very complex, and their target functions usually have multiple local optimums. The disadvantages of traditional optimization methods have led researchers to develop better optimization methods for solving real engineering optimization problems, thus developing heuristic optimization methods [1].

Lately, many metaheuristics algorithms have been successfully applied to solve a wide range of optimization problems [2–5]. The advantage of using these algorithms for solving complex problems is that they get optimal solutions even for problems of large dimensions in a short time [6]. A growing interest has been observed in metaheuristic methods over the last 2 decades. Authors in [6] gave the search result of the number of related studies for the metaheuristics on google scholar website (in May 2019). The algorithms are: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Tabu Search (TS), Genetic Programming (GP), Differential Evolution (DE), Ant Colony Optimization (ACO), Simulated Annealing (SA), Artificial Bee Colony (ABC), Greedy Randomized Adaptive Search Procedure (GRASP), Variable Neighbourhood Search (VNS), Firefly Algorithm (FA), Cuckoo Search (CS), Harmony Search (HS), Scatter Search (SS), Social Spider Optimization (SSO), Bacterial Foraging (BFO), Bat Algorithm (BA), Gravitational Search Algorithm (GSA) and Biogeography-based Optimization (BBO). GA and PSO have the largest numbers 1,270,000 and 263,000 related papers respectively.

Currently metaheuristic methods are successfully used to solve many problems of engineering optimization, such as multi-robot path planning, un-manned aerial vehicles navigation, the opinion leader detection in online social network, the identification of influential users in social network; the deployment of unmanned aerial vehicles, the data collection system of Internet of Things, the localization in wireless sensor network localization. In this article, we will consider the application of a metaheuristic algorithm, namely, the gravitational search algorithm, to solve the inverse problem of chemical kinetics in application to heterogeneous catalysis. In recent years, some metaheuristic optimization algorithms, such as Particle Swarm Optimization [7], Genetic Algorithms [8, 9], have been applied to solve chemical kinetics problems. However, there is still no persistent conclusion to select a certain algorithm to solve inverse problems of chemical kinetics.

This research is aimed at investigating the effectiveness of metaheuristic algorithms in solving inverse kinetic problems. This work is devoted to the implementation of a gravitational search algorithm for finding the kinetic parameters of an important industrial chemical process. GSA is one of the powerful metaheuristic

algorithms currently available that is utilized to solve numerous applications of optimization problems. Furthermore, researchers have proposed a large diversity of methods to improve GSA, such as using enhanced operators, hybridization of GSA with other heuristic algorithms, and parameter adaptation and control schemes for GSA. Despite the fact that GSA appeared recently, it is already widely used. The literature search showed that there is no work on applying the gravitational search algorithm to chemical kinetics problems now, which indicates the relevance of the research.

Mathematical model

The process under study is propane pre-reforming into methane-rich gas over Ni catalyst, which is an industrially important chemical process [10]. Pre-reforming of propane was studied over industrial nickel–chromium catalyst at pressure of 1 and 5 bar, low steam to carbon molar ratio of 1, in the temperature range of 220–380 °C and at flow rates of 4000 and 12,000 h^{-1}. The experimental data on propane pre-reforming was acquired from our previous work [11]. The reaction scheme consists of two reactions: propane steam conversion and CO_2 methanation [12]:

$$C_3H_8 + 6H_2O \rightarrow 10H_2 + 3CO_2 \tag{1}$$

$$CO_2 + 4H_2 \rightleftarrows CH_4 + 2H_2O \tag{2}$$

It was shown [11] that the Langmuir–Hinshelwood (LH) model correctly describes the experimental data and can be used to simulate the process of propane pre-reforming and predict propane conversion under given reaction conditions. So the reaction rates (1)–(2) are expressed according to the LH model:

$$W_{ref} = \frac{k^0_{ref} \cdot \exp\left(-\frac{E_{ref}}{RT}\right) \cdot C_{C3H8}}{\left(1 + B \cdot C_{C3H8}\right)^m};$$

$$W_{met} = k^0_{met} \cdot \exp\left(-\frac{E_{met}}{RT}\right) \cdot C_{H2} \cdot \left[1 - \frac{P_{CH4} P^2_{H2O}}{K_{eq} P_{CO2} P^4_{H2}}\right],$$

Here W_{ref} and W_{met} are the reaction rates; E_{ref} and E_{met} are the observed activation energies, J/mol; $k^0{}_{ref}$ and $k^0{}_{met}$ are the pre-exponential multipliers; B is the constant parameter, T is the temperature, K; R is the universal gas constant, J·(K·mol). The "*ref*" and "*met*" indexes refer to pre-reforming and methanation reactions, respectively. C_{C3H8} and C_{H2} are concentrations of propane and hydrogen, mol/m^3; m is order of the denominator, which varied from 0 to 2; K_{eq} is the equilibrium constant of CO_2 methanation; P_{CH4}, P_{H2O}, P_{CO2}, P_{H2} are partial pressures of the corresponding substances, bar. The mathematical model is a system of equations of material balance:

$$\begin{cases} G\frac{dy_i}{dl} = \left(\nu_i^{ref} W_{ref} + \nu_i^{met} W_{met}\right) m_i \\ 0 \le l \le L, \quad i \in \left\{C_3H_8, CH_4, H_2O, H_2, CO_2\right\} \\ l = 0, \quad y_i = y_{i0}, \end{cases} \tag{3}$$

Here G is a mass flow of the mixture, kg/(m^2 s); y_i is a mass fraction of the i-th component; ν_i is a stoichiometric coefficient of the i-th component; m_i is a molar mass of the i-th component, kg/mol; l is coordinate along the catalytic layer, m; L is a length of the catalytic layer, m. The length of the catalytic layer is 0.008 m. The mathematical model of chemical kinetics problems is a system of differential equations that describes the variations in substance concentrations over time according to the rates of reaction stages. The system of differential equations is a Cauchy problem containing the initial data [13–15]. The numerical solving of such a system of equations is a direct problem of chemical kinetics. Determining the kinetic parameters of reaction stages by comparing calculated values of substance concentrations and experimental results is an inverse problem of chemical kinetics. The mathematical problem is to minimize the functional of the deviation between calculated and experimental values. The functional of minimizations determined as the sum of absolute deviations between calculated and experimental concentrations:

$$F = \sum_{i=1}^{M} \sum_{j=1}^{N} \left| x_{ij}^{calc} - x_{ij}^{\exp} \right| \to \min, \tag{4}$$

Here x_{ij}^{calc} and x_{ij}^{exp} are calculated and experimental values of component concentrations; M is the number of measuring points; N is the number of substances involved in the reaction.

The framework of proposed algorithm

In this section, the algorithms used in this paper are briefly introduced. We consider Gravitational search algorithm for solving the inverse problems of chemical kinetics. GSA has been recently developed and found to be comparatively efficient. This algorithm is nature inspired and population-based. A brief description of the algorithm and pseudocode will be given.

Gravitational search algorithm (GSA)

GSA was proposed by Rashedi et al. [16]. In this algorithm, search agents are objects, and the best ones have a large mass. Objects are attracted to each other by the force of gravity. Agents with a large mass attract lighter ones. Each object represents a solution to the problem and has its own position, inertial mass, passive and active gravitational masses. The heaviest agent is the current best solution, and other agents are attracted by this agent. The GSA applies Newtonian laws of gravity and motion. Each object attracts each other, and the gravitational force between two

objects is proportional to the product of their masses and inversely proportional to the distance between them, R. That is in an environment with N objects the position of object i is equal to:

$$X_i = X_i^1,\ \ldots\ X_i^d,\ \ldots,\ X_i^n, \quad for\ i = 1, 2,\ \ldots,\ N,$$

Here X_i^d is the position of object i in the d^{th} dimension. The force applied to object "i" from agent "j" at time t is:

$$F_{ij}^d(t) = G(t)\frac{M_{pt}(t)M_{aj}(t)}{R_{aj}(t) + \varepsilon}\left(X_j^d(t) - X_i^d(t)\right),$$

Here M_{aj} is the gravitational mass applied to agent j, M_{pi} is the passive gravitational mass applied to agent i, $G(t)$ is gravitational at time t, ε is a small constant, and $R_{ij}(t)$ is the Euclidian distance between objects i ($i = 1, 2, \ldots, N$) and j ($j = 1, 2, \ldots, N$):

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2.$$

The total force that is applied to object i in d is a random sum of d^{th} components of the forces from other objects:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t),$$

Here $rand_j$ is a number in [0,1]. The acceleration of the object i at time t, and in direction d^{th}, $a_i^d(t)$ is given as:

$$a_i^d(t) = \frac{F_{ij}^d(t)}{M_{ii}^d(t)},$$

Here M_{ii} is the inertial mass of object i. The new velocity of an object is a fraction of its current velocity and its acceleration. Its position and velocity are calculates as follows:

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t),$$
$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1),$$

Here $rand_i$ is a uniform variable in [0,1]. The constant, G, is initialized and reduced with time to control the accuracy of the search:

$$G(t) = G\left(G_0, t\right). \tag{5}$$

The gravitational and inertial masses are updated by the equations given below:

$$M_{ai} = M_{pi} = M_{ii} = M_i; \quad i = 1,\ 2,\ \ldots,\ N;$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \quad M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)},$$

Here $fit_i(t)$ represents the fitness value of the agent i at time t, and, $worst(t)$ and $best(t)$ are defined as follows (for a minimization problem):

$$best(t) = \min_{j\in\{1,\ldots,N\}} fit(t), \quad worst(t) = \max_{j\in\{1,\ldots,N\}} fit(t).$$

The GSA pseudocode is shown in Table S1 (Supplementary Information) [6]. We developed a program in the Python 3 programming language, based on the code repository available at the link https://github.com/himanshuRepo/Gravitational-Search-Algorithm (open access).

Comparison of GSA with state-of-the-art algorithms

To evaluate the proposed GSA, five benchmark functions were used. All test functions have been listed in Table 1, where D indicates dimension of the function ($D = 30$), *Range* is the boundary of the function's search space and *Opt* is the global minimum. F_1–F_3 are unimodal functions, whereas F_4–F_5 are multimodal functions.

To compare the optimization performance among different algorithms, we used the next quality indicators—mean value and standard deviation. The smaller the mean value is, the stronger the global optimization ability of the algorithm is; the smaller the standard deviation is, the more stability the algorithm is. Table 2 shows the obtained results. In this table, $\bar{x}$ and s indicate "mean value" and "standard deviation" respectively. Best results are highlighted in bold. Mean value and standard deviation are calculated as follows:

Table 1 Benchmark functions

Formulation	Range	Opt
$F_1(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	$[-100,100]^{30}$	0
$F_2(\mathbf{x}) = \max_i \{\lvert x_i\rvert, 1 \le i \le D\}$	$[-100,100]^{30}$	0
$F_3(\mathbf{x}) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10,10]^{30}$	0
$F_4(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{D} \lvert x_i\rvert - \prod_{i=1}^{D} \cos\frac{x_i}{\sqrt{i}} + 1$	$[-600,600]^{30}$	0
$F_5(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{D} x_i^2}{D}}\right) - \exp\left(\frac{\sum_{i=1}^{D}\cos 2\pi x_i}{D}\right) + 20 + e$	$[-100,100]^{30}$	0

Table 2 The experimental results for five benchmark functions obtained by six metaheuristic algorithms

No.	Metric	CS	GWO	WOA	PSO	SSA	GSA
F_1	$\bar{x}$	**0**	**0**	**0**	**0**	4.81×10^{-9}	2.02×10^{-10}
	s	**0**	**0**	**0**	**0**	1.06×10^{-9}	1.06×10^{-9}
F_2	$\bar{x}$	16.6	**0**	12.8	3.24×10^{-5}	1.27	-6.06×10^{-11}
	s	4.13	**0**	19.4	3.59×10^{-5}	1.53	1.07×10^{-9}
F_3	$\bar{x}$	1.76×10^{-15}	**0**	**0**	**0**	5.24×10^{-1}	1.75×10^{-10}
	s	5.60×10^{-15}	**0**	**0**	**0**	7.33×10^{-1}	1.14×10^{-9}
F_4	$\bar{x}$	7.45×10^{-3}	**0**	4.14×10^{-4}	1.88×10^{-2}	6.73×10^{-3}	5.93×10^{-9}
	s	1.68×10^{-2}	**0**	2.27×10^{-3}	1.95×10^{-2}	7.17×10^{-3}	1.94×10^{-8}
F_5	$\bar{x}$	2	8.23×10^{-15}	3.02×10^{-15}	8.88×10^{-1}	2.18	-2.65×10^{-10}
	s	1.13	1.30×10^{-15}	2.21×10^{-15}	8.87×10^{-1}	5.88×10^{-1}	5.15×10^{-10}

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{D} x_i, \quad s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{D}\left(x_i - \bar{x}\right)^2}.$$

We compared the performance between GSA and five state-of-the-art algorithms: Cuckoo Search (CS), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), Particle Swarm Optimization (PSO) and Salp Swarm Algorithm (SSA) [1]. The statistical results obtained by six algorithms have been shown in Table 2. Despite the gravitational search algorithm did not show the best results, it appeared to be acceptable for our further work.

The following expression was used as a function of Eq. 5:

$$G = G_0 \exp\left(-\frac{\alpha t}{z}\right), \tag{6}$$

Here z is the number of iterations.

The parameters used in the GSA algorithm are as follows: population size = [100, 200], $\alpha = 30$, $G_0 = 250$. The convergence curves of the best objective functions value for benchmark functions F_1–F_5 are shown in Fig. 1. This figure proves that GSA can attain the best parameters in < 250 iterations.

After testing the algorithm on benchmark functions, the GSA was applied to solve a real problem, namely, the inverse problem for the process of propane pre-reforming into methane-rich gas over Ni catalyst (Fig. 2).

Simulation results

GSA for solving the inverse problem of chemical kinetics

In this section, the values of E_{ref}, E_{met}, k_{ref}, k_{met}, B and m, included in the expression of reaction rates W_{ref} and W_{met}, were optimized. When solving the inverse problem, the values of E_{ref}, E_{met}, k^0_{ref}, k^0_{met} varied, while the value of B varied from 0 to 5, and

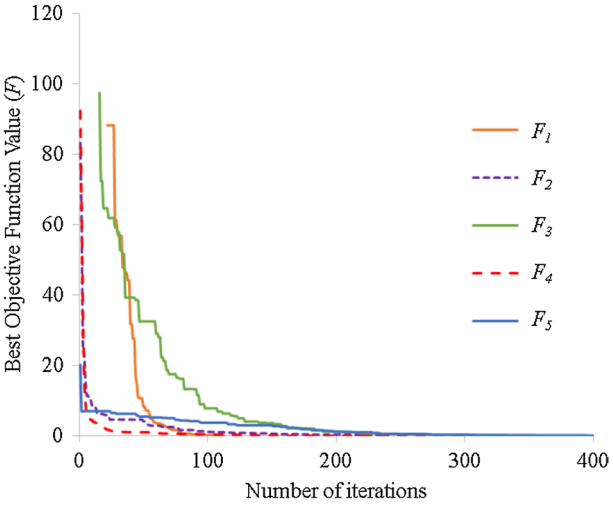

Fig. 1 The convergence curves for benchmark functions F_1–F_5, shown in Table 1. Dependence of the best objective function value on the number of iterations. The algorithm parameters are as follows: the population size is 100, $\alpha = 30$, $G_0 = 250$

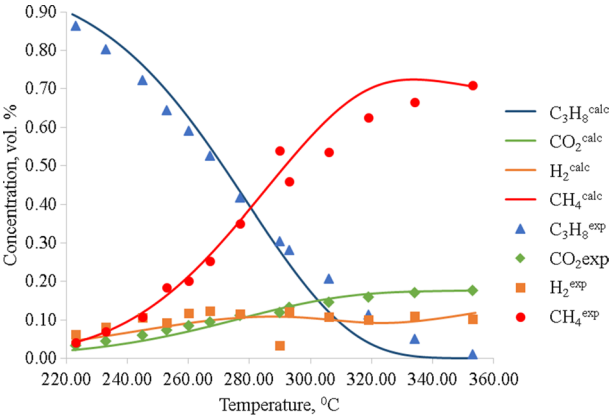

Fig. 2 Temperature dependences of the output concentrations of propane C_3H_8, methane CH_4, hydrogen H_2 and CO_2 in the process of propane pre-reforming. Experimental conditions: 220–380 °C, GHSV 4000 h^{-1}, 1 bar pressure, reaction mixture: 25 V% C_3H_8, 75 V% H_2O. Concentrations of the gas components on the figure are given on the dry basis. Points are experimental concentrations ("exp"-index), lines are simulated concentrations ("calc"-index)

m—from 0 to 2. The values obtained are shown in Table 3 in comparison with our previous work [11], in which we used a genetic algorithm (GA). The parameters of GSA were configured as follows: $\alpha = 30$, $G_0 = 250$, the population size was 1400. The value of the functional F has become slightly smaller in comparison with the genetic algorithm; the kinetic parameters have approximately the same order when

Table 3 The values E_{ref}, E_{met}, k_{ref}, k_{met}, B and n obtained as a result of solving the inverse problem

Method	E_{ref}, kJ/mol	E_{met}, kJ/mol	k_{ref}, $(mole/m^3)^m/s$	k_{met}, s^{-1}	m	B	F
GSA	103.1	39.7	10^{10}	4.7×10^4	0.9	0.37	0.024
GA [11]	105.1	44.2	1.3×10^{10}	1.6×10^5	1.1	0.20	0.025

solved by these two algorithms. Fig. 3 shows a comparison of rate coefficients that were achieved in this optimization and what were used in the earlier work [11] on an Arrhenius plot. The dependences of rate constant k_{ref} of the reaction 1 obtained by the GSA and GA algorithms are almost identical, while the dependences for the rate constant k_{met} of the reaction 2 are slightly different. This suggests that the kinetic parameters included in the W_{met} equation have a low sensitivity to the solution of this problem.

The obtained optimal values were used to solve the direct problem of chemical kinetics (Eq. 3). Fig. 4 shows the results of calculations for experiments on propane pre-reforming. The model correctly describes the available experimental data.

Parameter setting and sensitivity analysis

The sensitivity of some parameters of GS was experimentally investigated. A set of about 100 test problems (see Figs. 4, 5, 6) is utilized to study the sensitivity of parameters of GA. Each computational experiment was performed three times, the plots show the average value. The sensitivities of three important parameters such as population size, G_0 and α were studied. In the case of population size, 15 different values of population size in GSA, i.e. population size equal to 100, 200, …, 1500 are considered in this analysis. Values of G_0 and α were fixed ($G_0=250$, $\alpha=30$) and

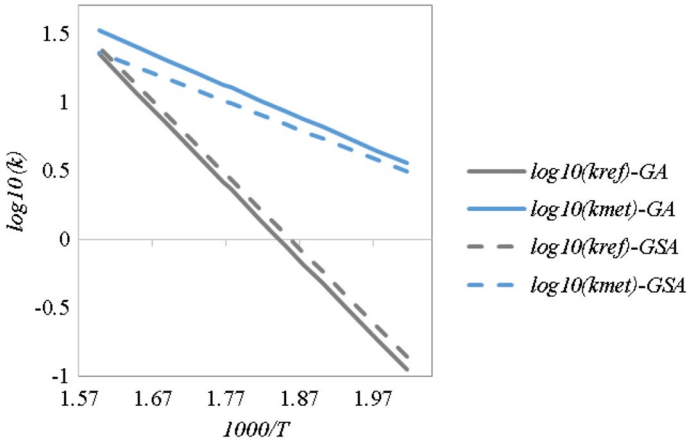

Fig. 3 Comparison of the rate constants of stages (1) and (2) obtained by the GA [11] and GSA algorithms on an Arrhenius plot

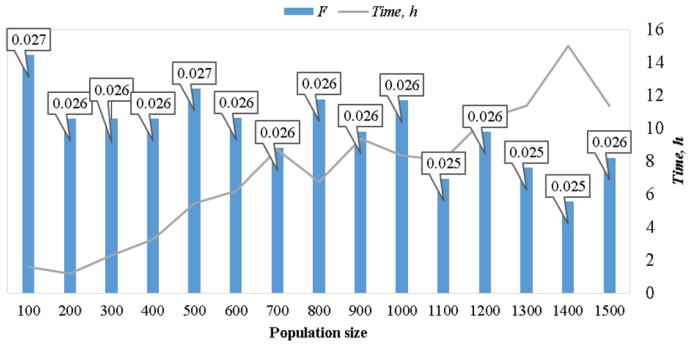

Fig. 4 Bars show the dependence of sum of absolute deviations between calculated and experimental concentrations (F in Eq. 4) on the population size. The line demonstrates the time spent on the calculation

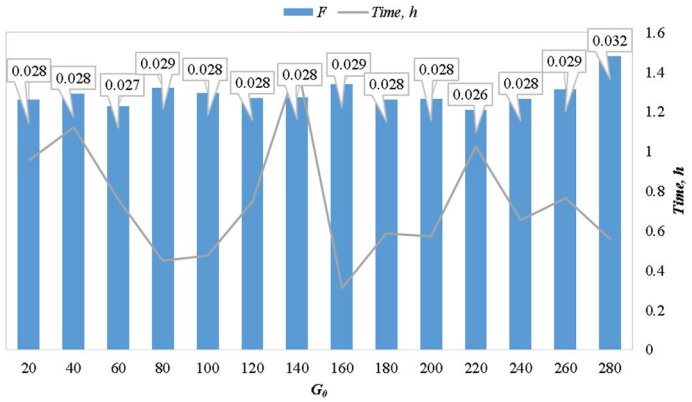

Fig. 5 Experimental results of GSA with varying the parameter G_0 from 20 to 280. Bars show the dependence of sum of absolute deviations between calculated and experimental concentrations (F in Eq. 4) on the parameter G_0. The line demonstrates the time spent on the calculation

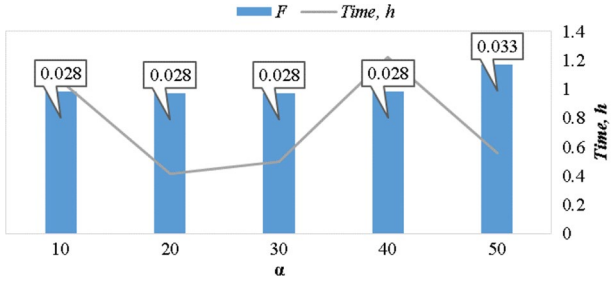

Fig. 6 Experimental results of GSA with varying α from 10 to 50. Bars show the dependence of sum of absolute deviations between calculated and experimental concentrations (F in Eq. 4) on the parameter α. The line demonstrates the time spent on the calculation

only the value of population size was modified in every simulation. The outcomes of functional F (see Eq. 4) and the time spent on the calculation are shown in Fig. 4. It is obvious that with increasing population size, the computational time increases. This is because the number of objects in the GSA environment increases and each object requires solving a system of ordinary differential equations (Eq. 3). However, Fig. 4 shows that when the population size is above 1400, the value of the functional F increases, therefore population size = [200; 1400] is recommended in this work.

For G_0, fifteen different values were considered, i.e. $G_0 = 10$, 20, 40, …, 280 to analyze the sensitivity of G_0 in the performance of the GSA. The other parameters were configured as follows: $\alpha = 30$, the population size is 100. The results are shown in Fig. 5. We have not found an explicit dependence of the value of the functional F and the calculation time on the value of the parameter G_0. The nature of the dependence of F on G_0 is not linear, so it can be concluded that the performance of GSA is not highly sensitive to value of G_0. We suggest using the G_0 parameter value of no more than 260.

We also investigated the effect of the parameter α in Eq. 6 on the convergence of the algorithm. The algorithm was run with variation of the parameter α in the range from 10 to 50 with a step of 10. The other parameters were configured as follows: $G_0 = 250$, the population size is 100 to speed up calculations, the number of iterations of the algorithm was 200. Fig. 6 shows the values of the functional F for different values of parameter α (bar chart) and the time spent on the calculation (line). The figure shows that for this problem, the optimal values of α are from 10 to 40, at which the minimum value of the functional F will be obtained in a relatively short time.

Conclusions

This paper shows that the gravitational search algorithm can be successfully applied to solve inverse problems of chemical kinetics. To show that the algorithm works, the algorithm has been tested on five benchmark functions, and the algorithm has shown good performance. After that, the gravitational search algorithm was used to find the kinetic parameters of the pre-reforming process. The obtained values of kinetic parameters coincide with those found in previous studies. Moreover, for this chemical process, we analyzed the sensitivity of the algorithm parameters to the problem solution. The ranges of algorithm parameter values that are most suitable for this task were found. The advantages of the algorithm include the following: ease of implementation, the ability to solve problems not only local, but also global optimization, what are the problems of chemical kinetics, as well as fast convergence of the algorithm. In the future, it is planned to apply this algorithm for kinetic modeling of other chemical processes.

Acknowledgements The reported study on optimization algorithms was funded by RFBR under the Project Numbers 19-37-60014 “Perspectiva” (Enikeeva L.V.). Experimental data were obtained with the support of the RFBR Project 19-33-60008 “Perspectiva” (Potemkin D.I.).

References

1. Yiying Z, Zhigang J (2020) Group teaching optimization algorithm: a novel metaheuristic method for solving global optimization problems. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113246
2. Shayanfar H, Gharehchopogh FS (2018) Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. Appl Soft Comput J. https://doi.org/10.1016/j.asoc.2018.07.033
3. Turgut OE, Turgut MS, Coban MT (2014) Design and economic investigation of shell and tube heat exchangers using Improved Intelligent Tuned Harmony Search algorithm. Ain Shams Eng J. https://doi.org/10.1016/j.asej.2014.05.007
4. Kaveh A (2014) Advances in metaheuristic algorithms for optimal design of structures. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-05549-7
5. Kumar A, Misra RK, Singh D, Mishra S, Das S (2019) The spherical search algorithm for bound-constrained global optimization problems. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.105734
6. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng. https://doi.org/10.1016/j.cie.2019.106040
7. Kazantsev KV, Bikmetova L, Dzhikiya O, Smolikov M, Belyi AS (2016) Particle swarm optimization for inverse chemical kinetics problem solving as applied to n-hexane isomerization on sulfated zirconia catalysts. Procedia Eng. https://doi.org/10.1016/j.proeng.2016.07.619
8. Gubaydullin I, Enikeeva L, Naik LR (2016) Software module of Mathematical Chemistry web-laboratory for studying the kinetics of oxidation of 4-tert-butyl-phenol by aqueous solution of H_2O_2 in the presence of titanosilicates. Eng J. https://doi.org/10.4186/ej.2016.20.5.263
9. Akhmadullina LF, Enikeeva LV, Gubaydullin IM (2017) Numerical methods for reaction kinetics parameters: identification of low-temperature propane conversion in the presence of methane. Procedia Eng. https://doi.org/10.1016/j.proeng.2017.09.654
10. Uskov SI, Potemkin DI, Kamboj N, Snytnikov PV, Pakharukova VP, Enikeeva LV, Gubaydullin IM, Hussainova I (2019) Fibrous alumina-based Ni-MOx (M = Mg, Cr, Ce) catalysts for propane pre-reforming. Mater Lett. https://doi.org/10.1016/j.matlet.2019.126741
11. Uskov SI, Potemkin DI, Enikeeva LV, Snytnikov PV, Gubaydullin IM, Sobyanin VA (2020) Propane pre-reforming into methane-rich gas over Ni catalyst: experiment and kinetics elucidation via genetic algorithm. Energies. https://doi.org/10.3390/en13133393
12. Uskov SI, Enikeeva LV, Potemkin DI, Belyaev VD, Snytnikov PV, Gubaidullin IM, Kirillov VA, Sobyanin VA (2017) Sobyanin kinetics of low-temperature steam reforming of propane in a methane excess on a Ni-based catalyst. Catal Ind. https://doi.org/10.1134/S2070050417020118
13. Enikeeva L, Gubaydullin I, Khursan S (2018) Numerical modeling of intramolecular transformations of ortho, meta-substituted aromatic nitroso oxides. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1096/1/012069
14. Enikeev MR, Potemkin DI, Enikeeva LV, Enikeev AR, Maleeva MA, Snytnikov PV, Gubaydullin IM (2020) Analysis of corrosion processes kinetics on the surface of metals. Chem Eng J. https://doi.org/10.1016/j.cej.2019.123131
15. Faskhutdinov AG, Faskhutdinova RI, Arefyev IA, Enikeeva LV (2019) Numerical simulation of the catalytic process of isomerization of pentane-hexane cut. J Phys Conf Ser. https://doi.org/10.1088/1742-6596/1368/4/042016
16. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci. https://doi.org/10.1016/j.ins.2009.03.004