

Article

Studying the Efficiency of Parallelization in Optimal Control of Multistage Chemical Reactions

Maxim Sakharov ^{1,*} , Kamila Koledina ², Irek Gubaydullin ² and Anatoly Karpenko ¹¹ CAD/CAE/PLM Department, Bauman Moscow State Technical University, 105005 Moscow, Russia² Institute of Petrochemistry and Catalysis of Russian Academy of Sciences, 450075 Ufa, Russia

* Correspondence: msakharov@bmstu.ru

Abstract: In this paper, we investigate the problem of optimal control of complex multistage chemical reactions, which is considered a nonlinear global constrained optimization problem. This class of problems is computationally expensive due to the inclusion of multiple parameters and requires parallel computing systems and algorithms to obtain a solution within a reasonable time. However, the efficiency of parallel algorithms can differ depending on the architecture of the computing system. One available approach to deal with this is the development of specialized optimization algorithms that consider not only problem-specific features but also peculiarities of a computing system in which the algorithms are launched. In this work, we developed a novel parallel population algorithm based on the mind evolutionary computation method. This algorithm is designed for desktop grids and works in synchronous and asynchronous modes. The algorithm and its software implementation were used to solve the problem of the catalytic reforming of gasoline and to study the parallelization efficiency. Results of the numerical experiments are presented in this paper.

Keywords: parallel computing; global optimization; desktop grids; optimal control; chemical reactions

MSC: 34H05; 34G20



Citation: Sakharov, M.; Koledina, K.; Gubaydullin, I.; Karpenko, A. Studying the Efficiency of Parallelization in Optimal Control of Multistage Chemical Reactions. *Mathematics* **2022**, *10*, 3589. <https://doi.org/10.3390/math10193589>

Academic Editors: Leonid B. Sokolinsky and Mikhail Zymbler

Received: 31 August 2022

Accepted: 26 September 2022

Published: 1 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many practical optimal control problems obtaining the optimal solution can be insufficient because they cannot be applied owing to various physical restrictions [1,2]. Additionally, modern systems that require any type of control are complex and usually described by high-dimensional systems of ordinary differential equations (ODEs) that have only numeric solutions [3,4]. As a result, achieving optimal control is impossible without robust numerical solvers [5].

In this paper, investigate an optimal control problem as a global constrained optimization task. A similar approach was used in [6,7] to control a wheeled robot and in [8] for gas allocation control, as well as for some other applications [9–11]. This type of problem belongs to a class of numeric methods that can obtain an approximate solution to the optimal control problem [12,13].

In this paper, we consider a complex chemical reaction involving the catalytic reforming of gasoline [14]. The specified process is of significant practical importance, as it produces commercial gasoline for daily use; therefore the reaction must be controlled to increase output of the target product and to reduce the output of undesired components. However, this process, like other complex multistage chemical reactions [15–17], involves many internal stages and multiple intermediate complexes, which makes the optimization task computationally expensive. Thus, in order to obtain a solution within a reasonable time, parallel computing systems are required [18]. However, the design of such a parallel optimization algorithm is a multifaceted task [19]. The algorithm should take into account not only problem-specific features but also peculiarities of the computing system in which

the algorithms are launched. In other words, an algorithm must be designed that is capable of both localizing suboptimal solutions within a reasonable number of iterations [20] and efficiently utilizing all available computing resources [18].

In this study, we consider the emerging class of loosely coupled computing systems, in particular, grid systems comprising heterogeneous personal computers (desktop grids), which are widely used for scientific computations [21,22]. When developing an optimization algorithm for this class of systems, communication expenses between computing nodes must be minimized. It is possible to achieve this task either with a peculiar optimization algorithm [23,24] or with a specific parallelization technique [22].

We propose a new parallel algorithm based on the mind evolutionary computation (MEC) algorithm [25] to solve an optimal control problem. The classical MEC algorithm appeared to be successful in solving real-world global optimization problems and to be suitable for parallelization [1,4,25], like similar population-based algorithms [24]. The proposed method takes the architecture of the desktop grid into account by minimizing the number of information exchanges between computing nodes and can work both in synchronous and asynchronous modes. In order to ensure the feasibility of solutions, the algorithm includes several strategies that consider various constraints imposed on the control parameters. For example, it helps to control the speed of change in control temperature or the change in component concentration, as well as the smoothness of the control function. These constraints are important for practical implementation of the obtained control strategies [26,27].

The remainder of this paper is organized as follows. Section 2 is devoted to problem formulation. In Section 3, the model of the gasoline catalytic reforming is described, and the problem of optimal control is formulated and transformed into a nonlinear global optimization problem. In Section 4, we propose synchronous and asynchronous parallel mind evolutionary computation algorithms. In Section 5, the results of numerical experiments are presented and analyzed both from the parallelization and chemical perspectives. In Section 6, we present our conclusions, summarize the study, and suggest directions for further work.

2. Problem Formulation

In this paper, we consider a deterministic global constrained nonlinear minimization problem:

$$\min_{X \in D \subset R^n} \Phi(X) = \Phi(X^*) = \Phi^*. \quad (1)$$

Here, $\Phi(X)$ is the scalar objective function, $\Phi(X^*) = \Phi^*$ is the required minimal value, $X = (x_1, x_2, \dots, x_n)$ is the n -dimensional vector of variables, R^n is the n -dimensional arithmetical space, and D is the constrained search domain.

Feasible domain D is determined with inequality constraints

$$D = \left\{ X \mid x_i^{\min} \leq x_i \leq x_i^{\max}, i \in [1 : n] \right\} \subset R^n. \quad (2)$$

Most chemical reactions within a class of problems under consideration are described by systems of ODEs [3,4]. However, it is difficult to find such a control function— $T(t)$, where t is the reaction time—that can provide the required output characteristics for a reaction, for instance, the maximum of the target product and/or the minimum of undesired substances.

In this work, the optimal control problem was transformed into a global optimization problem in the following manner. The integration interval $[t_{start}; t_{end}]$ is discretized so that the length of one section $[t_i; t_{i+1}]$ meets the restrictions imposed by the experimental unit. The values of $T(t_i)$, are the components of vector $X = (x_1 \dots x_n)$. A piecewise linear func-

tion is used for approximation of the function $T(t)$. The objective function is transformed into a function that minimizes the difference between current and required behavior:

$$J(T(t)) \rightarrow \min_{T(t)} \quad (3)$$

3. Optimal Control of the Catalytic Reformation of Gasoline

In this paper, we study a non-isothermal industrial reaction of the catalytic reformation of gasoline over a bimetallic catalyst ($Pt - Re/Al_2O_3$). The output of such a reaction is commercial gasoline for daily use.

During the catalytic reformation of gasoline, aromatic hydrocarbons are formed, which increase the octane number of the gasoline, which is the main purpose of the process. In addition, reformation is also used for the separate production of aromatic hydrocarbons, which are used in petrochemical processes. The octane number of gasoline determines its class and therefore its price.

Catalytic reforming is one of the main sources of aromatics in gasoline, along with catalytic cracking of vacuum gas oil and low-temperature catalytic isomerization of the pentane-hexane fraction. According to current standards, the main requirement is to reduce the proportion of benzol and aromatics in the gasoline composition. Therefore, one of the crucial tasks associated with the catalytic reforming of gasoline is to reduce the amount of aromatic hydrocarbons and benzol with minimal changes in the octane number [14].

The problem of aromatic hydrocarbon and benzol content, is solved as follows: pre-fractionation of raw materials (i.e., benzol-forming components are removed) or removal of excess aromatic hydrocarbons from commercial gasoline. In the first case, raw materials deteriorate, and productivity decreases [28]. In the latter case, the cost of the product increases. A possible solution to the above problem is the optimization of the reactor unit itself, which requires a detailed kinetic model of the process based on the fundamental laws of chemical transformations.

The target reactions of catalytic reforming are the reactions of formation of high-octane components: dehydrocyclization, dehydrogenation of naphthenes, isomerization of naphthenic, and dehydrogenation of naphthenic hydrocarbons [29–31]. Reactions that result in the splitting of a molecule into several smaller molecules are undesirable, as they form gases, which reduce the yield of the target reformat product. Target reactions have a total endothermic effect of about 200 kJ, and side reactions are exothermic.

The use of adiabatic reactors in the process affects the non-thermal nature of catalytic reforming. Therefore, in the kinetic model, it is necessary to take into account the change in temperature during the reaction in each adiabatic reactor.

In the kinetic model, individual hydrocarbons are represented as 37 groups, including normal paraffins (nP_i), isoparaffins (iP_i), five-membered naphthenes (ACP_i), six-membered naphthenes (ACH_i), and aromatic hydrocarbons (A_i), where i is the number of carbon atoms in the molecular structure, and hydrogen.

When developing a model for the catalytic reforming of gasoline, it is necessary to take into account the change in the number of molecules due to chemical transformations based on the scheme of transformations and grouping of individual components [16]. Because adiabatic reactors are used for reactions with heat absorption, the temperature of the mixture decreases along the catalyst bed (up to 80 °C), leading to a decrease in reaction rates. In the reforming model, it is necessary to describe the change in temperature over the catalyst bed.

Thus, the mathematical description of the non-isothermal reaction of the catalytic reforming of gasoline is expressed as (4)–(7). The mathematical model consists of 40 differential equations (group component equations, temperature change and moles) with initial data, with the following initial conditions: $y_i(0) = y_i^0$.

$$\frac{dy_i}{d\tau} = \sum_{j=1}^J v_{ij} w_j, \quad i = 1, \dots, I; \quad (4)$$

$$\frac{dT}{d\tau} = -\frac{\sum_{i=1}^I \frac{dy_i}{d\tau} \Delta H_i(T)}{\sum_{i=1}^I y_i C_{pi}(T)}$$

$$T(0) = T^0; \tag{5}$$

$$\Delta H_i(T) = \Delta H_i(298) + \int_{298}^T C_{pi}(T) dt$$

$$C_{pi}(T) = a_i + b_i T + c_i T^2 + d_i T^3 + e_i T^4;$$

$$\frac{dQ}{d\tau} = \sum_{i=1}^I \frac{dy_i}{d\tau}; Q(0) = Q^0; \tag{6}$$

$$w_j = k_j \cdot \prod_{i=1}^I \left(\frac{y_i}{Q}\right)^{|\alpha_{ij}|} - k_{-j} \cdot \prod_{i=1}^I \left(\frac{y_i}{Q}\right)^{\beta_{ij}};$$

$$k_j = k_j^0 \exp\left(-\frac{E_j}{RT}\right); \tag{7}$$

where y_i is the concentration of reaction reagents (mol/L); τ is the conditional contact time (kg·min/mol); J is the number of stages; I is the number of substances; v_{ij} is the coefficients of the stoichiometric matrix; w_j is the rate of the j -th stage (1/min); k_j and k_{-j} are the rate constants of direct and inverse reactions, respectively (1/min); α_{ij} is the negative elements of the matrix (v_{ij}); β_{ij} is the positive elements of the matrix (v_{ij}); k_j^0 and k_{-j}^0 are the pre-exponential factors (1/min); E_{j+} and E_{j-} are the activation energies of the direct and inverse reactions, respectively (kcal/mol); R is the gas contact (2 cal/(mol·K)); T is the temperature (K); t^* is the duration of the reaction (min); $\Delta H_i(T)$ is the formation enthalpy of the i -th component at temperature T (J/mol); $C_{pi}(T)$ is the specific thermal capacity of the i -th component at temperature T (J/(mol·K)); $a_i, b_i, c_i, d_i,$ and e_i are the coefficients of thermal capacity's temperature dependance of the i -th component; and Q is the mole discharge of the flow (mol/min).

The main challenges associated with the catalytic reforming of gasoline are restrictions on the content of benzol and the amount of aromatic hydrocarbons in the final reformat. As a result, a restriction is imposed on the introduction of the product of the process into commercial gasoline in terms of the content of these components. However, the main purpose of the catalytic reforming process is to increase the octane number (ON) of the gasoline. Therefore, along with environmental criteria, the necessary criterion of the optimal control task is to maintain the high value of the octane number of the mixture, which depends on the composition of the product.

The reactor block of the catalytic reforming process consists of three adiabatic reactors, each of which receives a mixture heated to the required temperature. The reactor block is represented by a cascade of successive adiabatic displacement reactors, $R1, R2,$ and $R3,$ filled with a catalyst. Before entering each of the reactors, raw materials are heated in ovens. Due to the strong endothermic effect of the reactions, as the reaction mixture passes through the catalyst layer, the temperature of the reaction mixture decreases, which negatively affects the rates of chemical reactions. In industry, the temperature at the inlet to each of the three reactors is kept within the range of 480–500 °C [32]. In this work, the following initial temperatures were used: $T_{R1}^0 = 489$ °C, $T_{R2}^0 = 489$ °C, and $T_{R3}^0 = 489$ °C.

The optimality criterion for the catalytic reforming of gasoline is based on the octane number of the reformat—the higher, the better.

$$Y_{ON}(T) = \sum_{i=1}^I y_i(T_1, T_2, T_3) \cdot ON_i \rightarrow max, \tag{8}$$

where ON_i is the octane number of the i -th component.

An increase in the octane number of gasoline during reformation is achieved due by the complete conversion of naphthenic (cyclic) hydrocarbons into arenes (aromatic hydrocarbons) with a high octane number, as well as by the partial conversion of paraffins into naphthenic hydrocarbons, which, in turn, are converted into arenes. However, according to environmental requirements, the contents of aromatic hydrocarbons $Y_A(T)$ and benzol

$Y_B(T)$ are limited to 40% and 2%, respectively. We treat these requirements as constraints in our optimization problem.

Another criterion with respect to the optimality of the catalytic reforming of gasoline is the yield of the target product, i.e., the reformate ($Y_R(T)$), which is calculated as the product of the process minus cracking gases. We use this criterion for reference only.

The resulting objective function and the constraints are presented below:

$$\begin{aligned} J(T(t)) &= -Y_{ON}(T(t_{end})) \rightarrow \min_{T(t)}; \\ Y_A(T(t_{end})) &< 0.4; \\ Y_B(T(t_{end})) &< 0.02. \end{aligned} \quad (9)$$

In this reaction, the temperature cannot be used as a control parameter, as it changes in each reactor. However, extra heating or cooling can be added to each reactor. In the model, this can be achieved by adding an extra summand to the equation that describes the change in temperature. In other words, we can control the first derivative ($T'_{extra}(t)$) or the speed of extra heating or cooling of each reactor. The working temperature range for the catalytic reforming of gasoline is approximately $T(t) \in [400 \text{ }^\circ\text{C}; 500 \text{ }^\circ\text{C}]$; considering the duration of this reaction and the initial temperatures for every reactor, the following constraints were determined: $T'_{extra}(t) \in [-0.002; 0.002]$. The piecewise linear function was selected for approximation of the function $T'_{extra}(t)$ on the interval $t \in [0; 89321]$ seconds. The reasonable time step differs slightly for each reactor and equals 240 seconds on average, thus the dimension of the optimization problem is $n = 378$.

4. Parallel Mind Evolutionary Computation Algorithms

In this work, we propose a novel parallel optimization algorithm based on the classical mind evolutionary computation algorithm [25]. The original MEC algorithm (simple MEC, SMEC) simulates some aspects of human behavior. An individual (s) is considered an intelligent agent that operates in a group (S) composed of analogous individuals. During the evolution process, every individual is affected by other individuals within the group. In order to achieve a high position within the group, an individual has to learn from the most successful individuals in the group, whereas groups should follow the same logic in intergroup competition.

The algorithm is composed of three main stages: initialization of groups, similar taxis, and dissimilation. Operations of similar taxis and dissimilation are repeated iteratively, while the best obtained value of an objective function ($\Phi(X)$) changes. When the best obtained value stops changing, the winner of the best group among a set of leading groups is selected as a solution to the optimization problem [33].

The SMEC algorithm can be interpreted as a multipopulation problem. A multipopulation consists of independent subpopulations with different instances of the SMEC algorithm. Each subpopulation is composed of leading groups ($S^b = (S_1^b, S_2^b, \dots, S_{|S^b|}^b)$) and lagging groups ($S^w = (S_1^w, S_2^w, \dots, S_{|S^w|}^w)$). The number of individuals within each group is set to be the same and equals $|S|$.

Every subpopulation has its own communication environment called a local blackboard, denoted as $C_k^l, k \in [1 : K]$, where K is the number of subpopulations. In addition, the whole multipopulation has a general global blackboard, i.e., C^g .

A multipopulation version of the SMEC algorithm can be used to decompose a problem and map it onto computing nodes of the loosely coupled system. In such a case, each subpopulation or group of subpopulations evolves independently on separate computational nodes. On the other hand, the absence of communication between nodes reduces the efficiency of optimization. In order to avoid this, different dynamic adaptation strategies should be added to the algorithm [34].

In this work, we incorporated the following changes into the multipopulation SMEC algorithm: a step adaptation strategy [35], projection onto a search domain technique to handle constraints [36], and a smoothing method to producing control strategies that can

be put into practice [17]. For the purpose of parallelization, the master–slave paradigm was used [18]. The description of the proposed parallel algorithm is presented below, and a flow chart diagram is displayed in Figure 1.

1. Initialization of groups within the search domain (D) and problem decomposition. This stage is performed on the master CPU.
 - a. Generate a given number (γ) of groups ($S_i, i \in [1 : \gamma]$); γ is the free parameter of the algorithm;
 - b. Generate a random vector ($X_{i,1}$). In this work, an LP_τ sequence was used [37]. Identify this vector with the individual ($s_{i,1}$) of the group (S_i);
 - c. For every $X_i, i \in [1 : \gamma]$, calculate the corresponding values of the objective function (Φ_i);
 - d. Randomly divide a set of vectors ($X_1, X_2, \dots, X_\gamma$) into K subpopulations, where K is the number of available computing nodes. The next steps are performed for each sub-population;
 - e. In every group ($S_{k,i}, k \in [1 : K], i \in [1 : \gamma/K]$), determine the initial coordinates of the rest of individuals in the group ($S_{k,i,j}, j \in [2 : |S|]$) according to the following formula:

$$X_{k,i,j} = X_{k,i,1} + N_{|X|}(0, \sigma). \tag{10}$$

Here, $|S|$ is the number of individuals in each group;

- f. Calculate the scores of all individuals in every subpopulation and put them on the corresponding local blackboards ($C_k, k \in [1 : K]$);
 - g. Create leading S^b and lagging S^w groups based on the obtained scores. The ratio between leading and lagging groups is determined by the free parameter η . In this work, the number of lagging and leading groups is the same ($\eta = 0.5$), corresponding to a situation in which there is a balance between exploration and exploitation.
2. The modified similar-taxis stage is launched independently on every parallel computing node and works with a subset of γ/K groups determined during the initialization stage. We describe the process for one computing node.
 - a. Take information on the current best individual ($s_{i,j}, j \in [1 : |S_i|]$) of the group (S_i) from the blackboard (C_k);
 - b. The value of parameter σ used to generate new agents decreases depending on the number of iterations:

$$\sigma = \begin{cases} \sigma_0, & \text{if } \lambda < \hat{\lambda}, \\ \frac{1}{(\lambda - \hat{\lambda})^\theta} + \varepsilon, & \text{if } \lambda \geq \hat{\lambda}. \end{cases} \tag{11}$$

Here, $\hat{\lambda}$ is the threshold number of iterations; when $\lambda \geq \hat{\lambda}$, the standard deviation (σ) begins to decrease; σ_0 is the initial value of the standard deviation; θ is the speed parameter (the recommended value $\theta = 0.2$); and ε is the tolerance used to identify the stagnation.

- c. Check for constraint violations:
 - i. If any component of vector X is outside of domain D , it is projected back on to the boundary of domain D [24]; and
 - ii. The difference between any two nearby components of vector X is modified in order not to overcome the specified limit from physical experiments. This procedure is performed in random order [3].
 - d. Create new leading groups ($S^b = (S^b_1, S^b_2, \dots, S^b_{|S^b|})$) and lagging groups ($S^w = (S^w_1, S^w_2, \dots, S^w_{|S^w|})$) around current best individuals ($\tilde{s}_{i,j}$) using Formula (10);

- e. Put information on the new winner on the corresponding local blackboard (C_k) and the global blackboard (C_g) available to the master.
3. The dissimilation stage is also performed on every parallel computing node.
 - a. Read the scores of all groups ($\Phi_i^b, \Phi_j^w, i \in [1 : |S^b|], j \in [1 : |S^w|]$) from the global blackboard (C^g);
 - b. Compare all the scores. If the score of any leading group (S_i^b) is less than score of any lagging group (S_j^w), then the lagging group becomes a leading groups, and the first group becomes a lagging group. If score of a lagging group (S_k^w) is lower than the scores of all leading groups for ω consecutive iterations, then it is removed from the population;
 - c. Each removed group is replaced with a new group via an initialization procedure.
4. For this study, the maximum allowed value of the objective function's evaluation λ_{max} was used as the termination criterion [36]. The synchronous approach was used, as the master node waits for the results or termination messages (via time out) from all computing nodes before composing the solution.

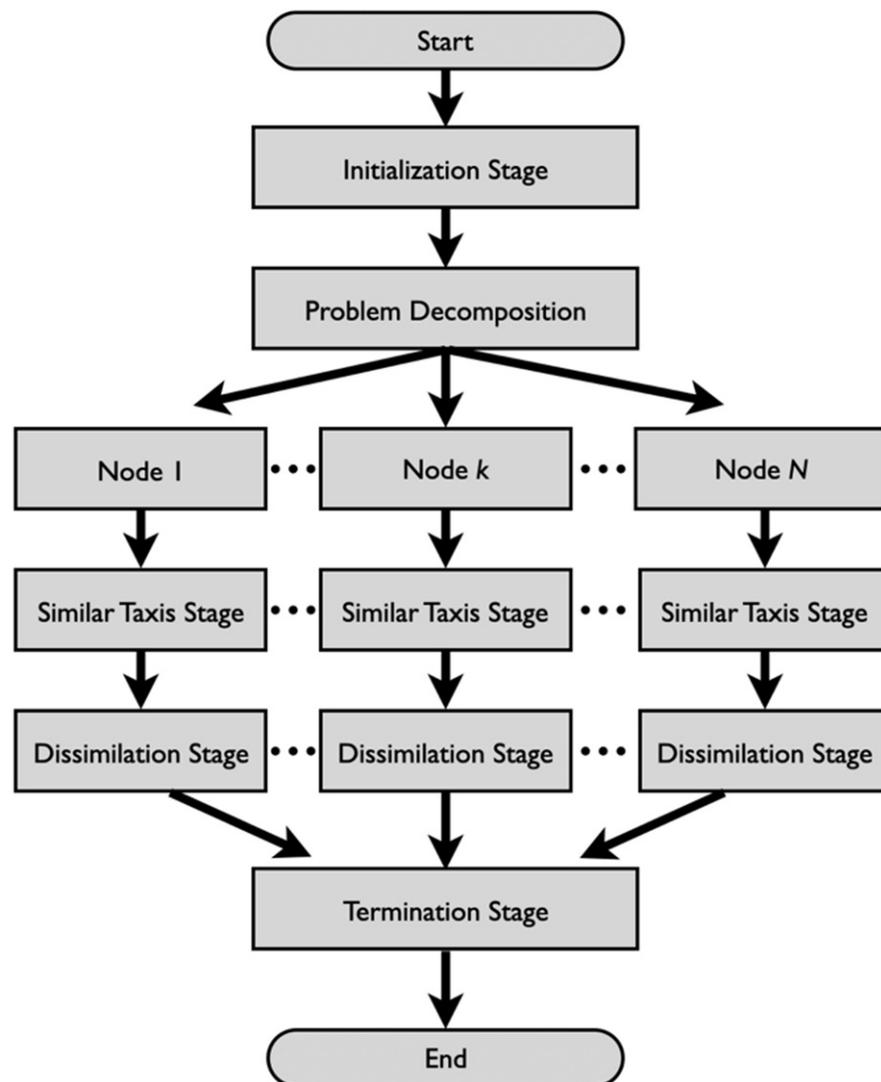


Figure 1. General flow chart of the parallel algorithm for synchronous and asynchronous versions of the PMEC algorithm.

Without any preliminary problem analysis and/or load-balancing procedure, the synchronous approach can be inefficient; some computational nodes can sit idle, whereas others still operate. As we focused on parallelization for desktop grids, dynamic load balancing would be impractical, as our goal was to minimize communication between the nodes.

Considering these aspects, we propose asynchronous parallel modification of the MEC algorithm. The idea of asynchronous learning was inspired by the [38], in which it was used for deep evolutionary reinforcement learning for large-scale problems. The idea is to update the evolutionary information every time a computing node finishes optimization.

The asynchronous update is performed by the master CPU after every new message received from the nodes and can be described as follows.

- a. Read the scores of all groups ($\Phi_i^b, \Phi_j^w, i \in [1 : |S^b|], j \in [1 : |S^w|]$) from the global blackboard (C^g);
- b. Compare all the scores. If score of any leading group (S_i^b) is less than the score of any lagging group (S_j^w), then the lagging group becomes a leading group, and the first group becomes a lagging group. If the score of a lagging group (S_k^w) is lower than scores of all leading groups for ω consecutive iterations, then it is removed from the population;
- c. Each removed group is replaced with a new group with via the initialization procedure;
- d. If a lagging group (S_j^w) was deleted while being processed on another computing node, its results are kept in order to form a new group as soon as another group is removed in order to maintain a constant number of groups and individuals and prevent information loss.

5. Computational Experiments and Analysis

Both synchronous parallel MEC (*sPMEC*) and asynchronous parallel MEC (*aPMEC*) algorithms were implemented using a combination of *Python* and *Wolfram Mathematica*. In addition, the sequential version of the modified algorithm was implemented for the efficiency study. Finally, we implemented another parallel version of the algorithm in which instead of problem decomposition, parallel computing nodes were utilized only for evaluation of the objective function (every individual is evaluated at the available node), whereas the general logic remained sequential. We refer to this algorithm as *oPMEC* (objective *PMEC*). It will be used only at the master node (multicore CPU), as it requires many information exchanges between nodes.

The specified problem of optimal control of the catalytic reforming of gasoline was used to compare the efficiency of the proposed parallelization techniques. For the experiments, a computational limit of 32,000 objective function evaluations was set for all four algorithms. The backward differentiation formula solver in *Wolfram Mathematica* was utilized to solve systems of ODEs. Computations for *sPMEC* and *aPMEC* were performed using a desktop grid composed of sixteen personal computers connected physically via a local network.

The following free-parameter values of the algorithm were utilized: the number of groups ($\gamma = 32$), the number of individuals in each group ($|S| = 15$), the removing frequency ($\omega = 30$ iterations), and the initial value of the standard deviation ($\sigma_0 = 0.0001$).

5.1. Study of the Parallelization Efficiency

A sequential experiment with one node was performed at the master node only, which is a personal computer with four CPU cores. For the specified computational limit, the sequential experiment took 159,217 s. In other words, one objective function evaluation takes approximately 5 s.

Parallelization with the *oPMEC* algorithm was also performed at the master node with four CPU cores ($N = 4$). In this work, we used the parallelization speedup (S) and the efficiency of parallelization (E) as two criteria for comparison of the algorithms [7]. The speedup is calculated as the ratio of program execution time on one node (T_1) to the

execution time on N nodes (T_N). The efficiency of the parallelization is calculated as the speedup (S_N) divided by the number of nodes (N). Because this type of parallelization does not involve any algorithm modification, its speedup and efficiency are poor (Figure 2); the exact values are presented in Table 1.

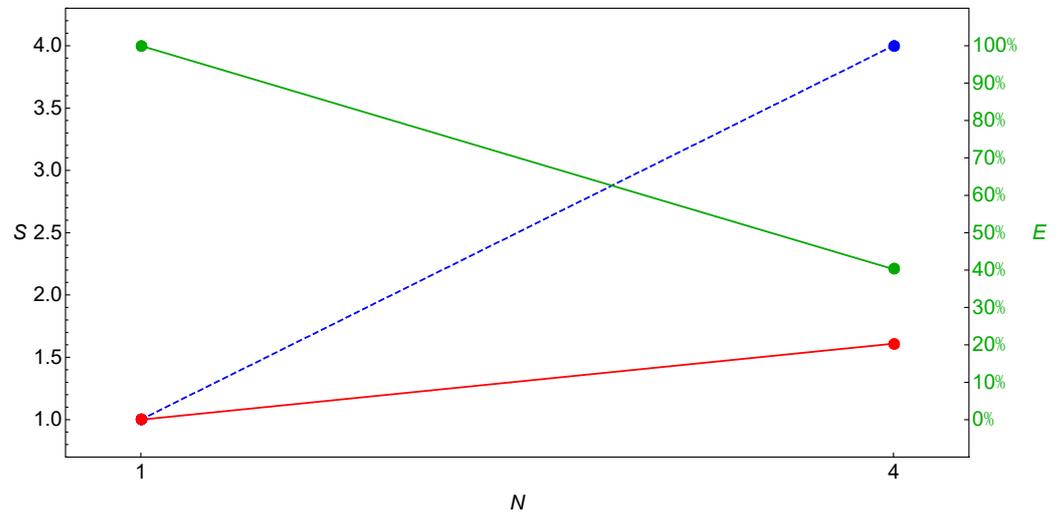


Figure 2. The speedup (S , red) and the efficiency (E , green) of the parallelization depending on the number of computing nodes (N) for the *oPMEC* algorithm; the maximum possible speedup is indicated by a blue dashed line.

Table 1. Speedup and parallelization efficiency for the *oPMEC* algorithm.

Number of Nodes, N	Speedup, S_N	Efficiency, E_N
4	1.61	40.3%

We also analyzed the *sPMEC* algorithm (Figure 3), which achieved relatively good speedup and efficiency when the number of nodes was low, although it scales up poorly. The parallelization efficiency is only 60% when the number of nodes is $N = 16$ (Table 2), mainly due to the absence of load balancing as a result of the loosely coupled architecture. As a result, some computing nodes took up to 75% more time to finish computations than others.

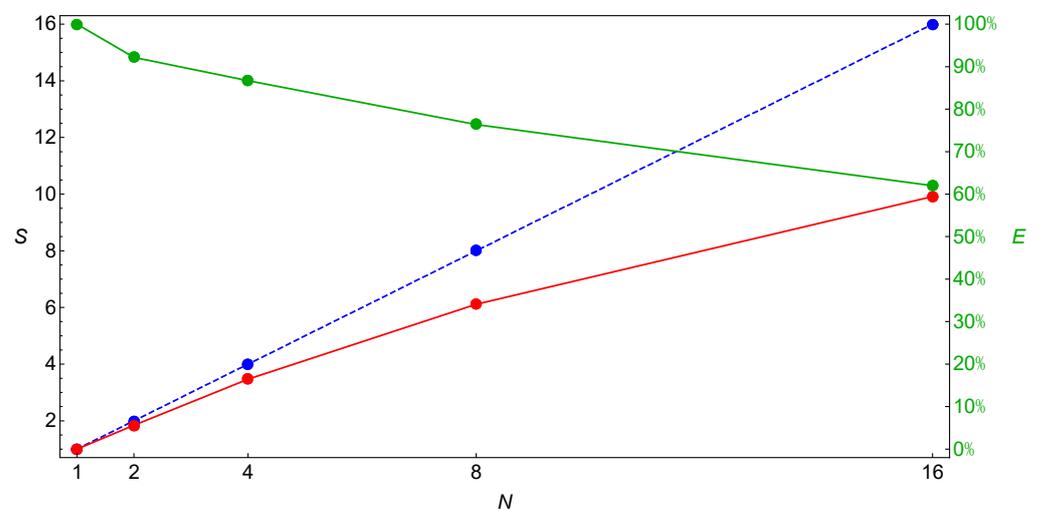
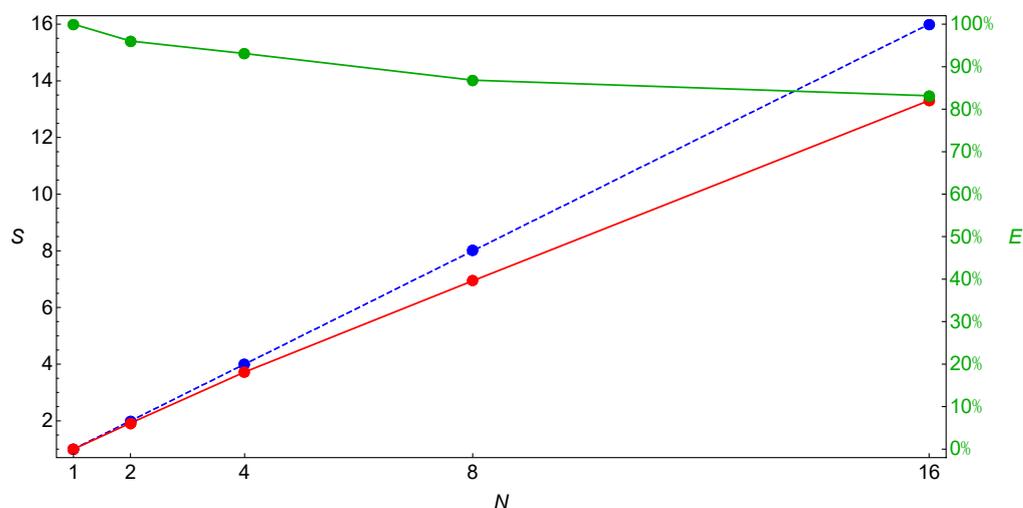


Figure 3. The speedup (S , red) and the efficiency (E , green) of the parallelization depending on the number of computing nodes (N) for the *sPMEC* algorithm; the maximum possible speedup is indicated by a blue dashed line.

Table 2. Speedup and parallelization efficiency for the *sPMEC* algorithm.

Number of Nodes, N	Speedup, S_N	Efficiency, E_N
2	1.84	92.2%
4	3.46	86.7%
8	6.11	76.4%
16	9.91	61.9%

Finally, we analyzed the *aPMEC* algorithm (Figure 4). This algorithm scales up better than the synchronous algorithm, resulting an efficiency of more than 80% when $N = 16$ (Table 3). Asynchronous evolution helped to avoid unnecessarily lengthy computation in the beginning of the process and resulted in high efficiency, as there were few steps performed at the master node.

**Figure 4.** The speedup (S , red) and the efficiency (E , green) of the parallelization depending on the number of computing nodes (N) for the *aPMEC* algorithm; the maximum possible speedup is indicated by a blue dashed line.**Table 3.** Speedup and parallelization efficiency for the *aPMEC* algorithm.

Number of Nodes, N	Speedup, S_N	Efficiency, E_N
2	1.92	96.0%
4	3.72	93.1%
8	6.95	86.8%
16	13.31	83.1%

5.2. Analysis from the Chemical Perspective

The introduction of temperature control throughout the process means determining the cooling regime to achieve the optimum according to the given criterion. In Figures 5 and 6, the vertical lines indicate the boundaries of the process by reactors (the reactor block of catalytic reforming consists of three reactors). The red dashed curves represent the scenario without $T'_{extra}(t)$, which is used as a baseline.

The cooling mode for the existing technological process shows the need for constant heat removal in the first and second reactors, as well as at the beginning of the third reactor; then, a decrease in cooling occurs, which is associated with the criterion of the optimal control problem to reduce the yield of aromatics. The authors of [28,29] showed that in the first and second reactors, as well as at the beginning of the third reactor, there is a constant accumulation of aromatic hydrocarbons at an elevated temperature; then, the content of aromatics decreases somewhat. Possible means of consumption include alkylation, hydrocyclization, and cracking reactions of alkyl substituents, which predominate in the

third reactor. The task of temperature control enables a reduction in the yield of aromatics to 36%, which is more than 20% less than without control (Table 4). The calculation is conducted as follows: $0.47 - 0.24 \times 0.47 = 0.47 - 0.11 = 0.36$; the calculation is conducted in a similar manner for all other rows in Table 4. Furthermore, the yield of benzene was reduced from 3 to 2%. Such a reduction in the yield of aromatic hydrocarbons and benzene allows the process product to comply with environmental restrictions. The value of *ON* decreases slightly, but it is not crucial for the quality of the commercial gasoline, as it also contains products of isomerization of the pentane-hexane fraction, which compensate for the drop. In turn, the aromatic hydrocarbons are generated only at this stage, and their decrease significantly improves the ecological properties of the gasoline. Furthermore, the yield of the reformat, i.e., the reaction product, increases from 86 to 93.

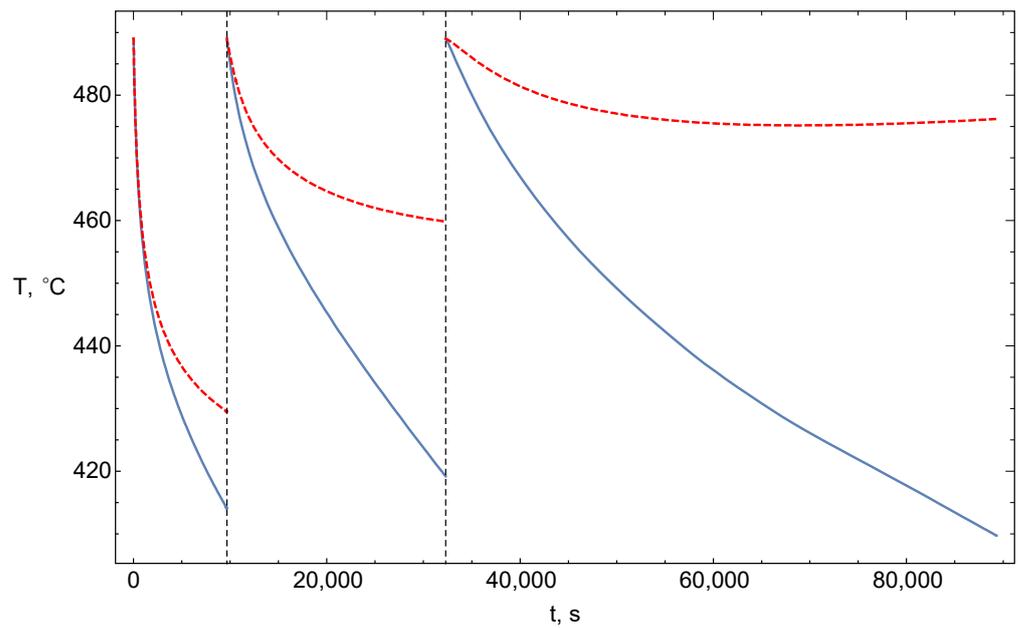


Figure 5. The reaction temperature in all three reactors for three different scenarios: red—no control; blue—obtained control; vertical dashed lines represent times in each reactor.

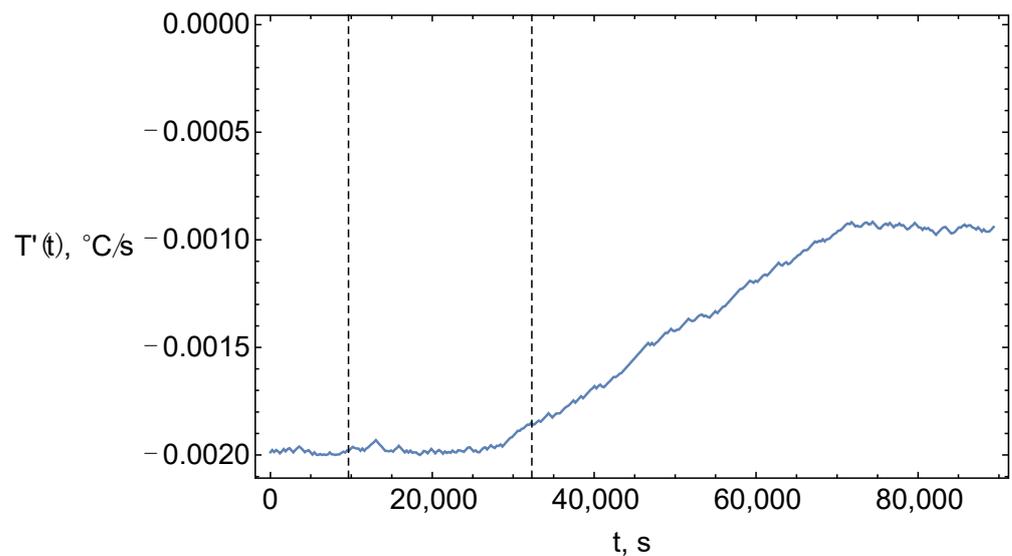


Figure 6. Obtained $T'_{extra}(t)$ for all three reactors *R1*, *R2*, and *R3*; vertical dashed lines represent times in each reactor.

Table 4. Target characteristics of the reaction with and without control.

Output Characteristic	without Control	with Obtained Control	Change in %
Octane Number	91.36	86.23	−6%
Target Product	86.57	93.74	+8%
Benzol	0.03	0.02	−33%
Aromatic Hydrocarbons	0.47	0.36	−24%

In general, the control of the thermal regime in reactors plays an important role, as the reaction rates change, which can lead to a change in the material balance and properties of the obtained products. Technological examples include when a heat exchanger and a refrigeration equipment are used in catalytic reforming units [39]. In the presented work, the optimal control of the catalytic reforming of gasoline is calculated, which makes it possible to provide specific technological recommendations with respect to operating conditions with heat removal in each reactor of the cascade throughout the entire process.

6. Conclusions

In this paper, we present a new parallel asynchronous MEC algorithm designed for optimal control of multistage catalytic reactions using desktop grids. Instead of a traditional evolution paradigm, the *aPMEC* algorithm operates in the asynchronous parallel mode, so learning is not required across the entire population before subsequent steps. Once local optimization is performed for any group, the master process can immediately perform another step of the algorithm. As a result, it rearranges computations to avoid down time in the beginning of the evolution process. The algorithm is also capable of finding feasible solutions by taking practical constraints into account.

The proposed algorithm and its software implementation were used to obtain feasible control for the catalytic reforming of gasoline. The results of the conducted numerical experiments and the obtained control are presented in this paper. The algorithm enabled improvement of the reaction characteristics. The yield of the reformate, i.e., the reaction target product, was increased from 86 to 93, and all environmental requirements were satisfied: the content of aromatic hydrocarbons was maintained under 40%, and that of benzol was maintained under 2%, from initial contents of 47% and 3%, respectively. Additionally, the parallelization efficiency was studied for the *aPMEC* algorithm and its synchronous variant. The asynchronous approach improved the parallelization efficiency for 16 computing nodes from 60% to 82%.

Future research will be devoted to investigating the performance of the proposed algorithm using hybridization with local search methods and other parallel architectures. Another possible research direction involves the concept of landscape analysis [24] and its importance for parallelization, as the results obtained during the landscape analysis stage can be used not only to tune the optimization algorithm but also for initial load balancing.

Author Contributions: Conceptualization, M.S. and K.K.; methodology, A.K.; software, M.S.; validation, M.S., K.K., I.G. and A.K.; formal analysis, M.S. and K.K.; investigation, M.S. and K.K.; resources, I.G.; data curation, K.K.; writing—original draft preparation, M.S.; writing—review and editing, M.S. and K.K.; visualization, M.S.; supervision, I.G. and A.K.; project administration, I.G.; funding acquisition, I.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant from the Russian Science Foundation (project No. 21-71-20047).

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewers for their valuable remarks on the content of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sakharov, M.; Karpenko, A. Parallel Multi-memetic Global Optimization Algorithm for Optimal Control of Polyarylenephthalide's Thermally-Stimulated Luminescence. In *Optimization of Complex Systems: Theory, Models, Algorithms and Applications*; WCGO 2019; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2020; Volume 991, pp. 191–201. [\[CrossRef\]](#)
2. Pontryagin, L.S.; Boltyanskii, V.G.; Gamkrelidze, R.V.; Mishchenko, E.F. *The Mathematical Theory of Optimal Processes*; Interscience: New York, NY, USA, 1962.
3. Rao, A.V. *A Survey of Numerical Methods for Optimal Control*; Preprint AAS 09-334; American Astronautical Society by Univelt: Escondido, CA, USA, 2015; pp. 1–32.
4. Sakharov, M.; Koledina, K.; Gubaydullin, I.; Karpenko, A. Optimal Control of Chemical Reactions with the Parallel Multi-Memetic Algorithm. In *Parallel Computational Technologies*; PCT 2020; Springer: Cham, Switzerland, 2020; Volume 1263, pp. 1–14. [\[CrossRef\]](#)
5. Frego, D.M. Numerical Methods for Optimal Control Problems with Application to Autonomous Vehicles. Ph.D. Thesis, University of Trento, Trento, Italy, 2014.
6. Diveev, A.I.; Konstantinov, S.V. Study of the practical convergence of evolutionary algorithms for the optimal program control of a wheeled robot. *J. Comput. Syst. Sci. Int.* **2018**, *57*, 561–580. [\[CrossRef\]](#)
7. Diveev, A.; Sofronova, E.; Shmalko, E. Modified SOMA for Optimal Control Problem. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation, (CEC 2019), Wellington, New Zealand, 10–13 June 2019; pp. 2894–2899.
8. Mahdiani, M.R.; Khamsehchi, E.; Suratgar, A.A. Using modern heuristic algorithms for optimal control of a gas lifted field. *J. Pet. Sci. Eng.* **2019**, *183*, 106348. [\[CrossRef\]](#)
9. Hager, W.W.; Pardalos, P.M. *Optimal Control: Theory, Algorithms and Applications. Applied Optimization*; Kluwer Academic Publishers: Norwell, MA, USA, 1998; 536p.
10. Diveev, A.I.; Konstantinov, S.V.; Sofronova, E.A. A Comparison of Evolutionary Algorithms and Gradient-based Methods for the Optimal Control Problem. In Proceedings of the 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT), Thessaloniki, Greece, 10–13 April 2018; pp. 259–264.
11. Yu, Y.; Zhang, C.; Gu, X.; Cui, Y. Expansion prediction of alkali aggregate reactivity-affected concrete structures using a hybrid soft computing method. *Neural Comput. Appl.* **2019**, *31*, 8641–8660. [\[CrossRef\]](#)
12. Fedorenko, R.P. *Approximate Solution of Optimal Control Problems*; Nauka: Moscow, Russian, 1978; 488p. (In Russian)
13. Tyatyushkin, A.; Zarodnyuk, T. Numerical method for solving optimal control problems with phase constraints. *Numer. Algebra Control Optim.* **2017**, *7*, 481–492. [\[CrossRef\]](#)
14. Stijepovic, M.Z.; Ostojic, A.V.; Milenkovic, I.; Linke, P. Development of a Kinetic Model for Catalytic Reforming of Naphtha and Parameter Estimation Using Industrial Plant Data. *Energy Fuels* **2009**, *23*, 979–983. [\[CrossRef\]](#)
15. Koledina, K.F.; Gubaidullin, I.M. Kinetics and mechanism of olefin catalytic hydroalumination by organoaluminum compounds. *J. Phys. Chem.* **2016**, *90*, 914–921. [\[CrossRef\]](#)
16. Bayguzina, A.R.; Gimaletdinova, L.I.; Khusnutdinov, R.I. Synthesis of Benzyl Alkyl Ethers by Intermolecular Dehydration of Benzyl Alcohol with Aliphatic Alcohols under the Effect of Copper Containing Catalysts. *Russ. J. Org. Chem.* **2018**, *54*, 1148–1155. [\[CrossRef\]](#)
17. Sakharov, M.; Koledina, K.; Gubaydullin, I.; Karpenko, A. Feasible Control of Chemical Reactions with the Parallel Mind Evolutionary Algorithm. In Proceedings of the Parallel Computing Technologies—XV International Conference, PaVT'2021, Volgograd, Russia, 30 March–1 April 2021; pp. 104–117.
18. Voevodin, V.V.; Voevodin, V.V. *Parallel Computations*; BHV: Saint-Petersburg, Russia, 2004; 608p.
19. Strongin, R.G.; Gergel, V.P.; Grishagin, V.A.; Barkalov, K.A. *Parallel Computing in the Global Optimization Problems*; Moscow State University Publishing House: Moscow, Russian, 2013; 280p. (In Russian)
20. Weise, T. *Global Optimization Algorithms—Theory and Application*; University of Kassel: Kassel, Germany, 2008; 758p.
21. Afanasiev, A.; Evtushenko, Y.; Posypkin, M. The layered software infrastructure for solving large-scale optimization problems on the grid. *Int. J. Comput. Res.* **2011**, *18*, 307.
22. Sakharov, M.; Karpenko, A. Comparative Study of a New Problem Decomposition Method for Solving Global Optimization Problems on Loosely Coupled Systems. In Proceedings of the Fifth International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'21); Kovalev, S., Tarassov, V., Snasel, V., Sukhanov, A., Eds.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2022; Volume 330. [\[CrossRef\]](#)
23. Sergeev, Y.D.; Grishagin, V.A. Parallel Asynchronous Global Search and the Nested Optimization Scheme. *J. Comput. Anal. Appl.* **2001**, *3*, 123–245.
24. Karpenko, A.; Agasiev, T.; Sakharov, M. Intellectualization Methods of Population Algorithms of Global Optimization. In *Cyber-Physical Systems: Advances in Design & Modelling*; Kravets, A., Bolshakov, A., Shcherbakov, M., Eds.; Studies in Systems, Decision and Control; Springer: Cham, Switzerland, 2020; Volume 259, pp. 137–151. [\[CrossRef\]](#)
25. Chengyi, S.; Yan, S.; Wanzhen, W. A Survey of MEC: 1998–2001. In Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics IEEE SMC2002, Hammamet, Tunisia, 6–9 October 2002; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2002; Volume 6, pp. 445–453. [\[CrossRef\]](#)
26. Diveev, A.; Sofronova, E.; Konstantinov, S. Approaches to Numerical Solution of Optimal Control Problem Using Evolutionary Computations. *Appl. Sci.* **2021**, *11*, 7096. [\[CrossRef\]](#)

27. Chertovskii, R.; Karamzin, D.; Khalil, N.T.; Pereira, F. Regular path-constrained time-optimal control problems in three-dimensional flow fields. *Eur. J. Control* **2020**, *56*, 98–106. [[CrossRef](#)]
28. Ismagilov, I.Z.; Matus, E.V.; Nefedova, D.V.; Kuznetsov, V.V.; Yashnik, S.A.; Kerzhentsev, M.A.; Ismagilov, Z.R. Effect of support modification on the physicochemical properties of a NiPd/Al₂O₃ catalyst for the autothermal reforming of methane. *Kinet. Catal.* **2015**, *56*, 394–402. [[CrossRef](#)]
29. Rozovskii, A.Y.; Kipnis, M.A.; Volnina, E.A.; Samokhin, P.V.; Lin, G.I. Selective CO oxidation on a Ru/Al₂O₃ catalyst in the surface ignition regime: 1. Fine purification of hydrogen-containing gases. *Kinet. Catal.* **2008**, *49*, 92–102. [[CrossRef](#)]
30. Zagoruiko, A.N.; Noskov, A.S.; Belyi, A.S.; Smolikov, M.D. Unsteady-state kinetic simulation of naphtha reforming and coke combustion processes in the fixed and moving catalyst beds. *Catalysis Today* **2014**, *220–222*, 168–177. [[CrossRef](#)]
31. Rahimpour, M.R.; Jafari, M.; Iranshahi, D. Progress in Catalytic Naphtha Reforming Process: A Review. *Appl. Energy* **2013**, *109*, 79–93. [[CrossRef](#)]
32. Zaynullin, R.Z.; Koledina, K.F.; Gubaydullin, I.M.; Akhmetov, A.F.; Koledin, S.N. Kinetic Model of Catalytic Gasoline Reforming with Consideration for Changes in the Reaction Volume and Thermodynamic Parameters. *Kinet. Catal.* **2020**, *61*, 613–622. [[CrossRef](#)]
33. Jie, J.; Zeng, J. Improved Mind Evolutionary Computation for Optimizations. In Proceedings of the 5th World Congress on Intelligent Control and Automation, Hang Zhou, China, 15–19 June 2004; pp. 2200–2204. [[CrossRef](#)]
34. Mersmann, O.; Bischl, B.; Trautmann, H.; Preuss, M.; Weihs, C.; Rudolph, G. Exploratory landscape analysis. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; ACM: Singapore, 2011; pp. 829–836. [[CrossRef](#)]
35. Sakharov, M.; Houllier, T.; Lépine, T. Mind Evolutionary Computation Co-algorithm for Optimizing Optical Systems. In Proceedings of the Fourth International Scientific Conference “Intelligent Information Technologies for Industry” (IITI'19), Ostrava–Prague, Czech Republic, 2–7 December 2019; Springer: Cham, Switzerland, 2020; pp. 476–486. [[CrossRef](#)]
36. Karpenko, A.P. Optimization Methods (Introductory Course). Available online: <http://bigor.bmstu.ru/> (accessed on 30 August 2022).
37. Sobol, I.M. On distribution of points in a cube and approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.* **1967**, *7*, 86–112. [[CrossRef](#)]
38. Gupta, A.; Savarese, S.; Ganguli, S.; Li, F.-F. Embodied Intelligence via Learning and Evolution. Available online: <https://arxiv.org/abs/2102.02202> (accessed on 30 August 2022).
39. Sukhanov, V.P. *Catalytic Processes in Oil Refining*; Kchimiya: Moscow, Russia, 1973; 373p.