UDC 004.94

METAHEURISTIC FRAMEWORKS FOR PARAMETER ESTIMATION IN APPROXIMATION MODELS

Grygor O. O. – Dr. Sc., Professor, Rector of the Cherkasy State Technological University, Cherkasy, Ukraine.

Fedorov E. E. – Dr. Sc., Professor, Professor of Department of Statistics and Applied Mathematics, Cherkasy State Technological University, Cherkasy, Ukraine.

Leshchenko M. M. – PhD, Associate Professor, Associate Professor of Department of International Economics and Business, Cherkasy State Technological University, Cherkasy, Ukraine.

Rudakov K. S. – PhD, Associate professor, Associate Professor of Department of Robotics and Specialized Computer Systems, Cherkasy State Technological University, Cherkasy, Ukraine.

Sakhno T. A. – PhD, Associate Professor of Department of International Economics and Business, Cherkasy State Technological University, Cherkasy, Ukraine.

ABSTRACT

Context. To enhance the performance of numerical optimization techniques, hybrid approaches integrating probabilistic modeling algorithms with annealing simulation have been introduced. These include Bayesian optimization, Markov-based strategies, and extended compact genetic algorithms, each augmented by annealing mechanisms. Such methods enable more precise search trajectories without requiring fitness function transformation, owing to their ability to explore the global search space in early iterations and refine the directionality of search in later stages.

Objective. The research aims to improve the effectiveness of parameter identification within approximation models of financial indicators by applying metaheuristic algorithms that incorporate probabilistic modeling and annealing-based simulation in intelligent computing systems.

Method. This study employs metaheuristic techniques grounded in probabilistic modeling and annealing-based simulation to enhance the accuracy and efficiency of parameter estimation within economic indicator approximation frameworks. Specifically, it introduces three hybrid strategies: Bayesian-based optimization integrated with annealing simulation, Markov-driven optimization enhanced by annealing, and an extended compact genetic algorithm coupled with annealing mechanisms. These methods enhance the accuracy of the search process by exploring the entire search space in initial iterations and refining the search direction in final iterations. The Bayesian optimization method employs a Bayesian network for structured search and solution refinement. The Markov optimization method integrates Gibbs quantization within a Markov network to improve search precision. The extended compact genetic algorithm utilizes limit distribution models to generate optimal solutions. These methods eliminate the need for fitness function transformation, optimizing computational efficiency. The proposed techniques expand the application of metaheuristics in intelligent economic computer systems.

Results. The implemented optimization strategies significantly enhanced the precision of parameter estimation within intelligent financial computing frameworks. The combination of probabilistic models and annealing simulation enhanced search efficiency without requiring fitness function transformation.

Conclusions. The proposed method expands the application of metaheuristics in economic modeling, increasing computational effectiveness. Further research should explore their implementation across diverse artificial intelligence problems.

KEYWORDS: probabilistic optimization frameworks, hybrid metaheuristic techniques, adaptive search algorithms, annealing-based simulation, computational parameter estimation, parametric identification, economics approximation model.

ABBREVIATIONS

GDP ss a gross domestic product;

ICT is an information and communications technology;

BOA is a Bayesian optimization algorithm;

MOA is a Markov optimization algorithm;

BIC is a Bayes information criterion;

BD is a Bayes-Dirichlet;

ECGA is an Extended Compact Genetic Algorithm;

MDL is a Minimum Descriptive Length.

NOMENCLATURE

A^{rp} is a reproduction operator;

A^{cr} is a probabilistic model creation operator;

A^{gn} is a descendant generation operator;

A^{rd} is a reduction operator;

N is a maximum number of iterations;

K is a population size;

$\widetilde{K}$ is a middle population size;

$\widehat{K}$ is a descendants' number;

$A^{\max}$ is an edges maximum count;

M is a length of chromosome;

E is bits count for gene of each chromosome;

$x_j^{\min}, x_j^{\max}$ are the minimum and maximum values for a chromosome;

x is a chromosome real vector;

b is a chromosome binary vector;

$U(0,1)$ is a function returning a standard uniformly distributed random value;

n is an iteration number;

G is a Bayesian network;

DOI 10.15588/1607-3274-2025-4-8

$\widetilde{G}$ is an ordered Bayesian network;

G_j^{in} is the set of vertices that preceding the vertex G_j^{num};

G_j^{out} is the set of vertices that following the vertex G_j^{num};

α is a significance parameter;

α_s is a natural parameter;

$equal()$ is a function returning the matched element count;

$round()$ is a function returning rounded number;

β is a specifying the cooling coefficient;

T_0 is an initial temperature;

$T^{\max}$ is a maximum temperature;

N_1 is an iterations maximum count;

N_2 is an iterations maximum count for Gibbs quantization;

P is a population;

$\widetilde{P}$ is a middle population;

$\hat{P}$ is a descendant's population;

$\mathbf{w}$ is a vector of weights;

p_{ij}, $p1_{ij}$, $p2_{ij}$ are a probability;

a^{avg} is a joint information matrix average value;

$S1$ is a disjoint set (set of clusters);

$S2$ is two different clusters set of unions;

$\mathbf{B}$ is a joint information symmetric matrix;

$C1_i^{\text{pop}}$, $C2_i^{\text{pop}}$ are a compressed population complexity;

$C1_i^{\text{model}}$, $C2_i^{\text{model}}$ are a model complexity;

$C1_i$, $C2_i$ are a combined complexity.

INTRODUCTION

The post-industrial era is marked by the expansion of electronic environments and the shift toward an information-driven society. In recent years, digitalization has emerged as a central challenge within the broader context of economic development, particularly in the information and communication technology sector. Digital transformation influences both economic progress and societal acceleration through complex, interconnected mechanisms, highlighting the need for standardized frameworks and reliable metrics to assess the digital economy and facilitate data exchange across industries. In this context, the implementation of parametric identification techniques for economic indicator approximation models – utilized in intelligent financial computing systems informed by digitalization metrics – has become increasingly significant. A key objective is to advance such identification methods to enhance model accuracy and system performance. Traditional random search algorithms lack convergence guarantees, while exact optimization techniques are computationally intensive. Local search-based methods, although more efficient, are prone to entrapment in local optima, underscoring the need for more effective optimization strategies.

Advanced heuristic approaches – commonly known as metaheuristics [1–3] – are utilized to expedite the search process for near-optimal solutions in numerical optimization tasks while reducing the likelihood of convergence to local extrema. These techniques extend classical heuristics by incorporating multiple strategies into a higher-order framework [4–6]. Population-based variants, in particular, are employed to enhance search speed and solution accuracy through collective and adaptive mechanisms [7–9]. The focus of this study is the parametric identification process applied to economic indicator approximation models. The research specifically investigates metaheuristic-based techniques designed to enhance the accuracy and efficiency of such identification methods.

The present research aims to enhance the efficiency of parametric identification applied to approximation models of economic indicators. To achieve this objective, a metaheuristic optimization method is proposed, integrating probabilistic modeling techniques with simulated annealing.

To fulfill the research goal, the following tasks were undertaken:

1) the development of a hybrid optimization method combining Bayesian optimization algorithms with annealing simulation;

2) the creation of a Markov-based optimization strategy supported by simulated annealing principles;

3) the formulation of an extended compact genetic algorithm integrated with annealing mechanisms; and

4) the execution of a numerical study to evaluate the proposed optimization techniques.

In parallel with these developments, contemporary trends reveal a growing prevalence of automated image analysis systems powered by machine learning algorithms. These technologies, widespread across domains such as diagnostics, remote sensing, and economic modeling, offer enhanced accuracy and adaptability, reinforcing the relevance and applicability of intelligent computational methods within financial systems.

1 PROBLEM STATEMENT

The core challenge addressed in this research involves enhancing the efficiency of solving numerical optimization tasks by employing probabilistic, multidimensional, and discrete evolutionary metaheuristic frameworks. Formally, the problem is defined as constructing a structured set of operators $\{A^{rp}, A^{cr}, A^{gn}, A^{rd}\}$, whose repeated application leads to a solution x^* satisfying the conditions $F(x^*) \to \min$ and $T \to \min$.

2 REVIEW OF THE LITERATURE

Existing metaheuristic frameworks have been observed to exhibit several limitations:

DOI 10.15588/1607-3274-2025-4-8

– many approaches provide descriptions tailored to specific problems or rely on abstract generalizations without broader applicability [10–12];

– they often lack support for non-binary candidate solution representations [13–15];

– the influence of iteration count on the search process is frequently overlooked [16–18];

– convergence to an optimal solution is not consistently guaranteed [19–21];

– there is limited capability to address constrained optimization problems [22–24];

– accuracy levels may be insufficient for complex or sensitive optimization tasks [25–27];

– parameter value determination procedures are typically non-automated and require manual tuning [28].

This highlights the ongoing challenge of developing robust and efficient metaheuristic optimization techniques. Among the widely adopted approaches are the Bayesian optimization algorithm [29], the Markov-based optimization framework [30], and the extended compact genetic algorithm [31], each offering distinct advantages in solving complex numerical problems.

3 MATERIALS AND METHODS

The extent of digital technology advancement across countries is primarily shaped by a set of indicators and influencing factors that collectively provide a multidimensional profile of each nation's socioeconomic landscape. These metrics serve to reflect both the degree of digital integration and the overall performance of digital systems within diverse national contexts.

A comprehensive dataset was compiled for the purpose of this study, encompassing macroeconomic indicators that reflect national economic performance – specifically, GDP per capita (final annual values in US dollars, denoted as Y). The dataset also integrates metrics related to digital technology development, which serve as explanatory variables. These include: X_1 – Frontier Technology Readiness Index (ICT component), annual; X_2 – Frontier Technology Readiness Index (Research and Development component), annual; X_3 – Share of ICT goods in total trade, annual percentage; X_4 – International trade in digitally deliverable services, annual value in millions of US dollars. All indicators were collected for the period spanning 2012 to 2022, covering 159 countries within the global economy.

The Bayesian optimization method incorporating annealing simulation was developed to enhance search efficiency and solution quality. The Bayesian Optimization Algorithm (BOA) [29], originally introduced by Pelikan, Goldberg, and Cantú-Paz [29], operates through a multiphase process: reproduction, probabilistic model construction via Bayesian networks, offspring generation based on the learned network, and a subsequent reduction phase. The underlying probabilistic graphical model employed in this approach is illustrated in Figure 1.

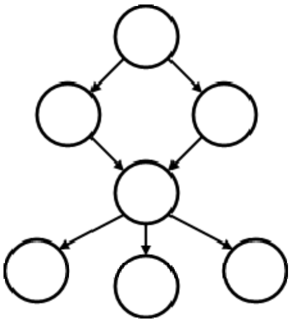

Figure 1 – Probabilistic BOA Graphical Model [29]

The methodology is structured into the following sequential phases:

1. Initial setup.

1.1. Defining the N, K, $\widetilde{K}$, $\widehat{K}$, $A^{\max}$ integrated into a specific vertex of the Bayesian structure: M, E, $x_j^{\min}, x_j^{\max}$, $j \in \overline{1,M}$.

1.2. Defining a fitness function $F(x) \to \min\limits_x$

1.3. Generation each source population chromosome k, $k \in \overline{1,K}$

a) generation each binary vector component j, $j \in \overline{1,M \cdot E}$

$\lambda = U(0,1)$.

$$b_{kj} = \begin{cases} 1, & \lambda < 0.5 \\ 0, & \lambda \ge 0.5 \end{cases}$$

b) creation of a real vector

$$x_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \frac{\sum\limits_{e=1}^{E} (2^{E-e} \cdot b_{k,(j-1)E+E+1-e})}{2^E - 1},$$

$j \in \overline{1,M}$.

c) when $(b_k, x_k) \notin P$, $P = P \cup \{(b_k, x_k)\}$.

1.4. Finding the best chromosome

$k^* = \arg\min\limits_k F(x_k)$, $x^* = x_{k^*}$

2. The count of iteration $n = 1$.

3. Replication and generation middle population $\widetilde{P}$.

3.1. To organize P by fitness function, $F(x_k) < F(x_{k+1})$.

3.2. A new population $\widetilde{P} = \{(\widetilde{b}_k, \widetilde{x}_k)\}$ of defined size $\widetilde{K}$ is generated through a hybrid approach that combines simulated annealing with linear order-based selection.

Calculation the choosing each i-th chromosome probability:

$$p(x_i) = \frac{1}{K} \exp(-1/T(n)) +$$
$$+ \frac{1}{K} \left(a - (2a-2) \frac{i-1}{K-1} \right) (1 - \exp(-1/T(n))),$$
$$T(n) = \beta T(n-1),\ 0 < \beta < 1,\ T(0) = T_0,\ T_0 > 0.$$

DOI 10.15588/1607-3274-2025-4-8

At both the initial and final phases of the genetic algorithm, distinct selection strategies are employed to balance exploration and exploitation. During the early iterations, equiprobable selection is applied, enabling broad coverage of the search space through random chromosome sampling. In contrast, the later stages utilize a linearly ranked selection mechanism that prioritizes high-performing chromosomes, thereby guiding the search toward promising regions. This dual-phase approach eliminates the need for fitness function transformation while maintaining search efficiency.

4. To develop a Bayesian network G.

4.1. Creating elements of Bayesian Network.

$$G_j = (G_j^{in}, G_j^{out}, G_j^{num}),$$

$G_j^{in} = \varnothing, G_j^{out} = \varnothing, G_j^{num} = j$, $j \in \overline{1, M \cdot E}$.

4.2. Directed connection in the Bayesian structure count $m = 1$.

4.3. $w^{\max} = -1$, $v^{from} = 0$, $v^{to} = 0$

4.4. The size of the vertex set defining the Bayesian network count $i = 1$.

4.5. Creation each viable descendant nodes j, $j \in \overline{1, M \cdot E}$, for the i-th parent node.

a) initialization of the visited vertices empty set $V^{viable} = \varnothing$, set a vertex j in the empty vertex stack S^V.

b) when the vertex i is on the S^V, $f^{pe} = 1$, move to stage j).

c) to pop a vertex from S^V and set it v^{cur}.

d) when $v^{cur} \in V^{visited}$, move to stage b)

e) $V^{visited} = V^{visited} \bigcup \{v^{cur}\}$.

f) when $|G_{v^{cur}}^{out}| = 0$, move to stage h)

g) for each l, $l \in \overline{1, |G_{v^{cur}}^{out}|}$, when $g_{v^{cur} l}^{out} \notin V^{visited}$, move S^V at the top $g_{v^{cur} l}^{out}$.

h) when the S^V is not empty, move to stage b)

i) $f^{pe} = 0$.

j) when $G_i^{num} \neq j \wedge j \notin G_i^{out} \wedge f^{pe} = 0$, then $V^{viable} = V^{viable} \bigcup \{j\}$.

4.6. To calculate each weights vector component j, $j \in \overline{1, M \cdot E}$, based on $K2$ or BIC or BD.

When $|G_j^{in}| = A^{\max}$ or $j \notin V^{viable}$, $w_j = -1$ then

a) $C = \{G_i^{num}\} \bigcup G_j^{in} \bigcup \{j\}$.

b) element sets $Q_1, ..., Q_{2^{|C|}}$ are derived, consisting of $\tilde{b}_k$ chromosomes, where the gene values are defined with values derived from the specified set C (that is, the values assigned to individual genes $\tilde{b}_{k,c_1}, ..., \tilde{b}_{k,c_{|C|}}$) match set Q_s with value s for $\tilde{b}_k$ chromosome was defined as $s = \sum_{e=1}^{|C|} 2^{|C|-e} \tilde{b}_{k,c_{E+1-e}}$, for chromosomes in Q_{2s-1}, Q_{2s} sets, the values assigned to individual genes from C set (genes values is $\tilde{b}_{k,c_1}, ..., \tilde{b}_{k,c_{|C|}}$) different corresponding to a single gene.

c) to assign a weight w_j to an edge (i, j), the following basis is applied:

– BD metric

$$w_j = \prod_{s=1}^{2^{|C|}/2} \frac{(\alpha_{2s-1} + \alpha_{2s} - 1)!}{(\alpha_{2s-1} - 1)!(\alpha_{2s} - 1)!} \cdot \frac{|Q_{2s-1} + \alpha_{2s-1} - 1|!|Q_{2s} + \alpha_{2s} - 1|!}{(|Q_{2s-1}| + |Q_{2s}| + \alpha_{2s-1} + \alpha_{2s} - 1)!},$$

when $\alpha_s = 1$, BD matches with $K2$.

– $K2$ metric

$$w_j = \prod_{s=1}^{2^{|C|}/2} \frac{|Q_{2s-1}|!|Q_{2s}|!}{(|Q_{2s-1}| + |Q_{2s}| + 1)!};$$

– BIC metric

$$w_j = \sum_{s=1}^{2^{|C|}/2} \left(|Q_{2s-1}| \ln \frac{|Q_{2s-1}|}{|Q_{2s-1}| + |Q_{2s}|} + \right. \left. + |Q_{2s}| \ln \frac{|Q_{2s}|}{|Q_{2s-1}| + |Q_{2s}|} \right) - $$
$$- \left(\frac{1}{2} \ln \tilde{K} \right) 2^{|C|}.$$

4.7. Determination a Best Edge (v^{from}, v^{to})

For each Bayesian network vertex count j, $j \in \overline{1, M \cdot E}$, when $w^{\max} < w_j$, then $w^{\max} = w_j$, $v^{from} = i$, $v^{to} = j$.

4.8. When $i < M \cdot E$, perform $i = i + 1$, move to stage 4.5.

4.9. When $w^{\max} > 0$, move to the most informative edge (v^{from}, v^{to}) for enhancing the Bayesian model: $G_{v^{to}}^{in} = v^{from}, G_{v^{from}}^{out} = v^{to}$.

4.10. When $m < A^{\max} \tilde{K}$, perform $m = m + 1$, move to stage 4.8.

5. Creation of a descendant $\tilde{P}$ population $\tilde{P}$ guided by the Bayesian network structure.

5.1. Generation Bayesian network $\tilde{G}$ with a defined topological sequence of nodes.

Integration of an additional element G_j into every Bayesian network structure $G_j^{count} = |G_j^{in}|$, $j \in \overline{1, M \cdot E}$ element. $\tilde{G} = \varnothing$, set a $\{G_j \,||\, G_j^{in}| = 0, j \in \overline{1, M \cdot E}\}$ subset

DOI 10.15588/1607-3274-2025-4-8

of elements in the empty S^G. Create each element q, $q \in \overline{1, M \cdot E}$, for G_j component:

a) detaching S^G element from the $\tilde{G}_q$ and set it.

b) when $|\tilde{G}_q^{out}| = 0$, move to stage h).

c) $l = 1$.

d) $j^* = \underset{j}{equl}(G_j^{num}, \tilde{g}_l^{out})$.

e) $G_{j^*}^{count} = G_{j^*}^{count} - 1$.

f) when $G_{j^*}^{count} \le 0$, to detach element S^G from the G_{j^*}.

g) when $l < |\tilde{G}_q^{out}|$, perform $l = l + 1$, move to stage d).

h) $\tilde{G} = \tilde{G} \bigcup \{\tilde{G}_q\}$.

5.2. Creation each descendant population chromosome i, $i \in \overline{1, K}$, using PLS method.

a) Generation each binary vector component j, $j \in \overline{1, M \cdot E}$. Child vector is initialized as set like $\widehat{b}_i = (0,...,0)$.

When $|\tilde{G}_j^{in}| = 0$, to determinate the probability $p_j = \frac{1}{\tilde{K}} \sum_{k=1}^{\tilde{K}} \tilde{b}_{kj}$, to generate a binary vector $\widehat{b}_{ij} = \begin{cases} 1, & p_j < 0.5 \\ 0, & p_j \ge 0.5 \end{cases}$ and move to stage b).

$C = \{G_j^{num}\} \bigcup G_j^{in}$.

Sets $Q_1, ..., Q_{2^{|C|}}$ are generated, consisting of chromosomes $\tilde{b}_k$, in which the genes values with values from the C (genes values are $\tilde{b}_{k,c_1}, ..., \tilde{b}_{k,c_{|C|}}$) represent a valid match, and the set Q_s value s for the chromosome $\tilde{b}_k$ is specified like $s = \sum_{e=1}^{|C|} 2^{|C|-e} \tilde{b}_{k,c_{E+1-e}}$, for chromosomes in the Q_{2s-1}, Q_{2s} sets, values assigned to genes with values from C set (genes values are $\tilde{b}_{k,c_1}, ..., \tilde{b}_{k,c_{|C|}}$) different exclusively for individual gene.

$C1 = G_j^{in}$.

To compute $p_j = \frac{|Q_a|}{|Q_a| + |Q_b|}$,

$$a = \sum_{e=1}^{|C1|} 2^{|C1|-e} \widehat{b}_{i,c1_{E+1-e}} + 2^{|C1|},\ b = \sum_{e=1}^{|C1|} 2^{|C1|-e} \widehat{b}_{i,c1_{E+1-e}},$$

for chromosomes in Q_a, Q_b sets, value of the genes with numbers in the C set (i.e., the genes values is $\tilde{b}_{k,c_1}, ..., \tilde{b}_{k,c_{|C|}}$) different exclusively for $\tilde{b}_{k,c_1}$ gene associated with the given vertex G_j^{num}.

To generate a binary vector

$$\widehat{b}_{ij} = \begin{cases} 1, & p_j < 0.5 \\ 0, & p_j \ge 0.5 \end{cases}$$

b) generating a real vector

$$\widehat{x}_{ij} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \frac{\sum_{e=1}^{E} (2^{E-e} \cdot \widehat{b}_{i,(j-1)E+E+1-e})}{2^E - 1},$$

$j \in \overline{1, M}$.

c) $\widehat{P} = \widehat{P} \bigcup \{(\widehat{b}_i, \widehat{x}_i)\}$.

6. Reduction.

6.1. Merging descendants $\widehat{P}$ and population P

6.2. Ordering union by fitness function, $F(x_k) < F(x_{k+1})$.

6.3. Constructing a refined population P using the most optimal chromosomes K identified in the aggregated set.

7. To calculate the best chromosome

$k^* = \arg\min_k F(x_k)$

8. When $F(x_{k^*}) < F(x^*)$, perform $x^* = x_{k^*}$.

9. When $n < N$, then $n = n + 1$, move to stage 3.

The result is x^*.

The Markov Optimization Algorithm (MOA), incorporating simulated annealing, was developed to enhance search efficiency and solution quality. Originally introduced by Shakya and Santana [30], this method operates through a structured sequence of phases: reproduction, construction of a probabilistic Markov network, generation of offspring guided by the network and Gibbs sampling principles, and a final reduction phase. The underlying probabilistic graphical model utilized in this approach is illustrated in Figure 2.

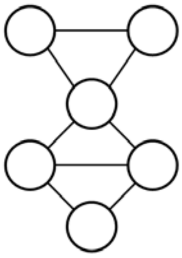

Figure 2 – Probabilistic structure of the Markov Optimization Algorithm [30]

The algorithmic workflow is delineated through the following sequence:

1. Initialization.

1.1. Determining the β, $T^{\max}$, α, and $0 < \beta < 1$, $T^{\max} > 0$, $\alpha > 1$.

DOI 10.15588/1607-3274-2025-4-8

1.2. Setting the N_1, N_2, K, $\tilde{K}$, $\widehat{K}$, $A^{\max}$ connecting a Markov network subset of vertices with one vertex; M, E, $x_j^{\min}, x_j^{\max}$, $j \in \overline{1, M}$.

1.3. Determining the fitness function $F(x) \to \min\limits_x$.

1.4. Generation each source population chromosome k, $k \in \overline{1, K}$,

a) generation each binary vector component j, $j \in \overline{1, M \cdot E}$

$$\lambda = U(0,1).$$

$$b_{kj} = \begin{cases} 1, & \lambda < 0.5 \\ 0, & \lambda \geq 0.5 \end{cases}$$

b) generation a real vector.

$$x_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \frac{\sum\limits_{e=1}^{E} (2^{E-e} \cdot b_{k,(j-1)E+E+1-e})}{2^E - 1},$$

$j \in \overline{1, M}$.

c) when $(b_k, x_k) \notin \tilde{P}$, perform $P = P \cup \{(b_k, x_k)\}$

1.5. Finding the best chromosome $k^* = \arg\min\limits_k F(x_k)$, $x^* = x_{k^*}$.

2. Iteration count $n = 1$.

3. Replication and generation middle population $\tilde{P}$.

3.1. To sort P by fitness function, i.e. $F(x_k) < F(x_{k+1})$.

3.2. To generate $\tilde{P} = \{(\tilde{b}_k, \tilde{x}_k)\}$ population $\tilde{K}$ size from a P population based on a simulated annealing with combination of linearly ordered selection and random selection.

Calculation the choosing each i-th chromosome probability:

$$p(x_i) = \frac{1}{K} \exp(-1/T(n)) +$$
$$+ \frac{1}{K} \left(a - (2a - 2) \frac{i-1}{K-1} \right) (1 - \exp(-1/T(n)))$$

$$T(n) = \beta T(n-1),\ 0 < \beta < 1,\ T(0) = T_0,\ T_0 > 0.$$

This mechanism is applied at both the initial and terminal phases of the genetic algorithm's operation. In its early phase, the algorithm employs uniform random selection to enable broad exploration of the search space through stochastic sampling of chromosomes. In the final phase, a linearly ranked selection strategy is adopted, shifting the process toward targeted exploitation by retaining the currently best-performing individuals. This integrated approach operates without requiring modification of the fitness function.

4. To generate a Markov network G.

4.1. Adjustment of a component j, within the unconditional probability vector $j \in \overline{1, M \cdot E}$

a) a pair of chromosomes $\tilde{b}_k$ sets $Q0, Q1$ is constructed, where gene values g_j are consistent across members, and for these sets $Q0$ $\tilde{b}_{kj} = 0$, $Q1$ $\tilde{b}_{kj} = 1$.

b) $p(g_j = 0) = \dfrac{|Q0|}{\tilde{K}}$, $p(g_j = 1) = \dfrac{|Q1|}{\tilde{K}}$.

4.2. Modification of each element (i, j), within the joint probability vector: $i \in \overline{1, M \cdot E - 2}$, $j \in \overline{2, M \cdot E - 1}$.

a) the sets $Q00, Q01, Q10, Q11$ are formed, containing the chromosomes $\tilde{b}_k$, where the j-th gene values and the i-th gene values coincide; for $Q00$ $\tilde{b}_{kj} = 0, \tilde{b}_{ki} = 0$, for $Q01$ $\tilde{b}_{kj} = 0, \tilde{b}_{ki} = 1$, for $Q10$ $\tilde{b}_{kj} = 1, \tilde{b}_{ki} = 0$, for $Q11$ $\tilde{b}_{kj} = 1, \tilde{b}_{ki} = 1$

b) $p(g_i = 0, g_j = 0) = \dfrac{|Q00|}{\tilde{K}}$,

$$p(g_i = 1, g_j = 0) = \frac{|Q01|}{\tilde{K}},$$

$$p(g_i = 0, g_j = 1) = \frac{|Q10|}{\tilde{K}},$$

$$p(g_i = 1, g_j = 1) = \frac{|Q11|}{\tilde{K}}.$$

4.3. Generating of each joint information symmetric matrix element (i, j), $i \in \overline{1, M \cdot E - 2}$, $j \in \overline{2, M \cdot E - 1}$ when $i < j$, perform

$$b_{ij} = \sum_{g_i = 0}^{1} \sum_{g_j = 0}^{1} p(g_i, g_j) \ln\left(\frac{p(g_i, g_j)}{p(g_i) p(g_j)} \right),\ b_{ij} = b_{ji}.$$

4.4. Calculation of the joint information matrix average value:

$$a^{avg} = \frac{2}{(M \cdot E - 1)^2 - (M \cdot E - 1)} \sum_{i=1}^{M \cdot E - 2} \sum_{\substack{j=2 \\ j>i}}^{M \cdot E - 1} b_{ij}.$$

4.5. To expand the Markov network by adding new links (i, j) between nodes

$$i \in \overline{1, M \cdot E - 2},\ j \in \overline{2, M \cdot E - 1},$$

when $i < j$ and $b_{ij} > a^{avg} \alpha$, to add an edge (i, j) to the Markov network.

4.6. Deleting worst edges for each vertex i, $i \in \overline{1, M \cdot E - 2}$

If a node i is connected to more than a specified number $A^{\max}$ of other nodes, retain only the top edges $A^{\max}$ with the highest values b_{ij}.

4.7. To determine adjacent elements for each vertex j, $j \in \overline{1, M \cdot E}$

To generate surroundings U_j from vertices linked by edges to a vertex j.

DOI 10.15588/1607-3274-2025-4-8

5. To generate each descendants' population chromosome k, $k\in\overline{1,K}$, from a population $\widetilde{P}$ based on a Gibbs quantization and Markov network.

5.1. To initialize every entry j of the binary vector, $j\in\overline{1,M\cdot E}$

$$\lambda=U(0,1).$$

$$\widehat{b}_{kj}=\begin{cases}1, & \lambda<0.5\\ 0, & \lambda\geq 0.5.\end{cases}$$

5.2. During each iteration of Gibbs-based quantization l, $l\in\overline{1,N_2}$

a) random selection of the index corresponding to the modified gene, i.e., $j=round(1+(M-1)U(0,1))$, $g_j=\widehat{b}_{kj}$.

b) two groups of chromosomes $Q0,Q1$ are formed, where the gene values $\widetilde{b}_m$ from the surrounding context U_j match those within the corresponding positions $\widehat{b}_k$ of each chromosome, and for $Q0$ $\widehat{b}_{kj}=0$, for $Q1$ $\widehat{b}_{kj}=1$.

c) $p(g_j=0\,|\,U_j)=\dfrac{|Q0|}{\widetilde{K}}$, $p(g_j=1\,|\,U_j)=\dfrac{|Q1|}{\widetilde{K}}$.

d) $\lambda=U(0,1)$.

e) changing of the j-th gene value of a binary vector:

$$b_{kj}=\\=\begin{cases}0, & \lambda<\dfrac{\exp\left(\dfrac{p(g_j=0)}{T(l)}\right)}{\exp\left(\dfrac{p(g_j=0\,|\,U_j)}{T(l)}\right)+\exp\left(\dfrac{p(g_j=1\,|\,U_j)}{T(l)}\right)}\\ 1, & \lambda\geq\dfrac{\exp\left(\dfrac{p(g_j=0)}{T(l)}\right)}{\exp\left(\dfrac{p(g_j=0\,|\,U_j)}{T(l)}\right)+\exp\left(\dfrac{p(g_j=1\,|\,U_j)}{T(l)}\right)}.\end{cases}$$

f) $T(l+1)=\beta T(l)$, where $T(1)=T^{\max}$

5.3. To generate a real vector:

$$\widehat{x}_{kj}=x_j^{\min}+(x_j^{\max}-x_j^{\min})\frac{\sum_{e=1}^{E}(2^{E-e}\cdot\widehat{b}_{k,(j-1)E+E+1-e})}{2^E-1},$$

$j\in\overline{1,M}$.

5.4. $\widehat{P}=\widehat{P}\bigcup\{(\widehat{b}_k,\widehat{x}_k)\}$.

6. Reduction.

6.1. Integrate descendants $\widehat{P}$ into the current population P.

6.2. To sort union by fitness function, i.e., $F(x_k)<F(x_{k+1})$.

6.3. Finding the K first (best) chromosomes in the union and generation a new population P.

7. Finding the best chromosome $k^*=\arg\min\limits_k F(x_k)$.

8. When $F(x_{k^*})<F(x^*)$, $x^*=x_{k^*}$.

9. When $n<N_1$, perform $n=n+1$, move to stage 3.

The result is x^*.

The extended compact genetic algorithm (ECGA) [31], enhanced with simulated annealing, was developed to improve optimization performance in complex search spaces. Originally introduced by Harik [31], ECGA operates through a structured sequence of stages: reproduction, construction of a probabilistic model based on the product of marginal limit distributions, offspring generation guided by this model, and a final reduction phase. The probabilistic graphical representation underlying this method is depicted in Figure 3.

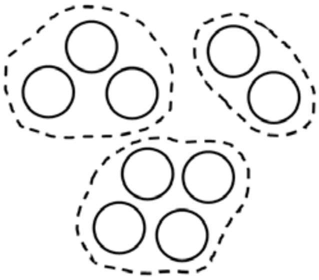

Figure 3 – The probabilistic model underlying the ECGA optimization approach [31]

The developed method unfolds through the following sequential steps:

1. Initialization.

1.1. To set N, K, $\widetilde{K}$, $\widehat{K}$, M, E, $x_j^{\min},x_j^{\max}$, $j\in\overline{1,M}$.

1.2. Specification of the fitness evaluation function $F(x)\to\min\limits_x$.

1.3. Generation each source population chromosome k, $k\in\overline{1,K}$

a) generation each binary vector j, $j\in\overline{1,M\cdot E}$

$\lambda=U(0,1)$

$$b_{kj}=\begin{cases}1, & \lambda<0.5\\ 0, & \lambda\geq 0.5\end{cases}$$

b) generation a real vector:

$$x_{kj}=x_j^{\min}+(x_j^{\max}-x_j^{\min})\frac{\sum_{e=1}^{E}(2^{E-e}\cdot b_{k,(j-1)E+E+1-e})}{2^E-1},$$

$j\in\overline{1,M}$.

c) when $(b_k,x_k)\notin\widetilde{P}$, perform $P=P\bigcup\{(b_k,x_k)\}$.

1.4. Finding the best chromosome $k^*=\arg\min\limits_k F(x_k)$, $x^*=x_{k^*}$.

2. Count of iteration $n=1$.

DOI 10.15588/1607-3274-2025-4-8

3. Replication and generation middle population $\widetilde{P}$.

3.1. Sort P according to goal function $F(x_k) < F(x_{k+1})$.

3.2. Population $\widetilde{K}$ size is regenerated $\widetilde{P} = \{(\widetilde{b}_k, \widetilde{x}_k)\}$ through a hybrid mechanism combining simulated annealing with both deterministic ranking and random selection techniques.

Calculation the choosing each i-th chromosome probability:

$$p(x_i) = \frac{1}{K}\exp(-1/T(n)) +$$

$$+\frac{1}{K}\left(a - (2a-2)\frac{i-1}{K-1}\right)(1 - \exp(-1/T(n))),$$

$T(n) = \beta T(n-1)$, $0 < \beta < 1$, $T(0) = T_0$, $T_0 > 0$.

4. Generation of a probabilistic model.

4.1. Generation of a disjoint sets (set of clusters) $S1$ containing one gene (vertex), i.e., $S1 = \{S1_j\}$, $S1_j = \{j\}$, $j \in \overline{1, M}$

4.2. Calculation combined complexity for each cluster i, $i \in \overline{1, |S1|}$ (used the MDL metric).

a) estimating computational intricacy $C1_i^{\text{model}} = \log_2 \widetilde{K}(2^{|S1_i|} - 1)$.

b) chromosome subsets $Q_{i1},...,Q_{i|S1|}$ are formed from the intermediate population, with gene values aggregated according to their cluster-specific frequencies $S1_i$ (i.e., the gene value $\widetilde{b}_{k,s1_{ij}}$) is equal to 1

c) probability computation $p1_{ij} = \dfrac{|Q_{ij}|}{\widetilde{K}}$.

d) evaluation of population encoding efficiency:

$$C1_i^{pop} = \widetilde{K}\sum_{j=1}^{2^{|S1_i|}} - p1_{ij}\log_2 p1_{ij}\ ,\ i \in \overline{1, |S1|}.$$

e) calculating the combined complexity (used the MDL metric) $C1_i = C1_i^{\text{model}} + C1_i^{\text{pop}}$.

4.3. Generating a set of two differ clusters unions $S2$.

a) $q = 1$, $S2 = \varnothing$

b) for each pair vertex (i, j), $i \in \overline{1, |S1|-1}$, $j \in \overline{1, |S1|}$, if $i < j$, then $S2_q = S1_i \bigcup S1_j$, $i_q = i$, $j_q = j$, $q = q+1$.

4.4. calculation combined complexity for each combined cluster i, $i \in \overline{1, |S2|}$ (used the MDL metric).

a) computing model complexity $C2_i^{model} = \log_2 \widetilde{K}(2^{|S2_i|} - 1)$.

b) population-derived chromosome sets $Q_{i1},...,Q_{i|S2|}$ are formed, embedding gene values alongside their set-specific frequencies $S2_i$ (the gene value $\widetilde{b}_{k,s2_{ij}}$) is equal to 1

c) probability assessment $p2_{ij} = \dfrac{|Q_{ij}|}{\widetilde{K}}$.

d) evaluation of execution complexity $C2_i^{pop} = \widetilde{K}\sum\limits_{j=1}^{2^{|S2_i|}} - p2_{ij}\log_2 p2_{ij}$.

e) estimation of total model complexity using the MDL principle $C2_i = C2_i^{model} + C2_i^{pop}$.

4.5. Finding the best cluster union

$$q^* = \arg\max_q (C1_{i_q} + C1_{j_q} - C2_q),$$

$$\Delta C^{best} = C1_{i_{q^*}} + C1_{j_{q^*}} - C2_{q^*}.$$

4.6. If $\Delta C^{best} > -1$, then $S1 = (S1 \bigcup \{S2_{q^*}\}) \setminus (\{S1_{i_{q^*}}\} \bigcup \{S1_{j_{q^*}}\})$, move to stage 4.2.

5. To generate each descendants' population chromosome k, $k \in \overline{1, K}$, from a population $\widetilde{P}$ based on the generate marginal distributions model

a) generation of binary vectors for each cluster i, $i \in \overline{1, |S1|}$

Random initialization of a vector d populated with chromosome frequency values $\{1,..., \widehat{K}\}$

$$\widehat{b}_{ks1_{ij}} = \widetilde{b}_{d_k s1_{ij}}\ ,\ k \in \overline{1, \widehat{K}}\ ,\ j \in \overline{1, S1_i}\ .$$

b) initialization of real-valued feature vectors

$$\widehat{x}_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min})\frac{\sum\limits_{e=1}^{E}(2^{E-e} \cdot \widehat{b}_{k,(j-1)E+E+1-e})}{2^E - 1},$$

$j \in \overline{1, M}$, $k \in \overline{1, K}$

c) $\widehat{P} = \{(\widehat{b}_k, \widehat{x}_k)\}$.

6. Reduction.

6.1. Merge newly generated descendants $\widehat{P}$ with the current population P

6.2. Ranking the combined population based on fitness scores, i.e., $F(x_k) < F(x_{k+1})$.

6.3. Finding the K first (best) chromosomes in the union and generation a new population P.

7. Identifying the optimal individual based on fitness evaluation

$$k^* = \arg\min_k F(x_k).$$

8. When $F(x_{k^*}) < F(x^*)$, perform $x^* = x_{k^*}$.

9. When $n < N$, perform $n = n+1$, move to stage 3.

The result is x^*.

4 EXPERIMENTS

Experimental simulations were conducted using the scikit-opt Python module, which provides a flexible framework for implementing metaheuristic optimization

DOI 10.15588/1607-3274-2025-4-8

algorithms. The Pandas library was employed for data preprocessing, including linear interpolation to address missing values and tabular input/output operations.

The study utilized a comprehensive database of economic indicators (https://unctadstat.unctad.org/wds/ReportFolders/reportFolders.aspx?sCS_ChosenLang=en) covering 159 countries over a 10-year period (2012–2022). The dataset includes both macroeconomic performance metrics and digital development indicators. Key parameters used in the optimization process were as follows: population size: $K = 100$, maximum number of iterations: $N = 100$, initial temperature: $T_0 = 106$, cooling coefficient: $\beta = 0.94$, middle population size: $\widetilde{K} = 100$, descendants number: $\widehat{K} = 100$, edges maximum number $A^{\max} = 2$, $a = 1.7$.

5 REZULTS

The cooling behavior during annealing is modeled using a temperature decay equation $T(n) = \beta^n T_0$, graphically represented in Figure 4.

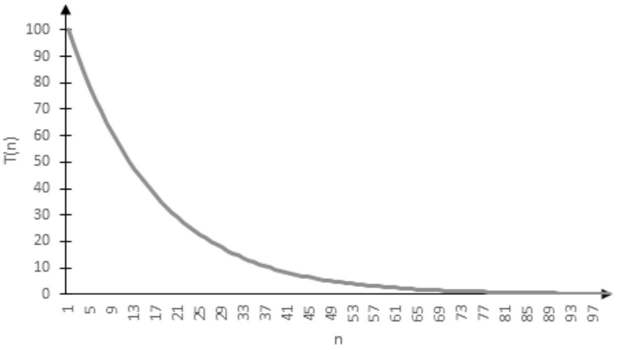

Figure 4 – Visualization of the annealing cooling strategy

The relationship between annealing temperature and iteration count, as illustrated in Figure 4, demonstrates a gradual decline in temperature as the number of iterations increases.

A comparative analysis of the proposed optimization techniques versus classical approaches is summarized in Table 1.

Table 1 – RMSE-Based evaluation of proposed and classical optimization algorithms

№	Method	Mean squared error evaluation of the proposed method	
		Proposed annealing-based algorithm	traditional technique
1	Bayesian optimization algorithm	0.04	0.08
2	Markovian optimization algorithm	0.03	0.07
3	Extended compact genetic algorithm	0.02	0.06

6 DISCUSSIONS

The parameter settings adopted in the proposed numerical optimization techniques ensure a dynamic mutation strategy: a high mutation probability is maintained during the initial iterations to promote broad exploration, while a gradual reduction in mutation probability during later stages facilitates convergence toward optimal solutions.

In contrast, classical probabilistic model-based methods typically overlook the influence of iteration count within the reproduction operator. This omission can lead to reduced solution accuracy, as evidenced by the comparative results presented in Table 1. The newly developed approaches effectively address these limitations by incorporating adaptive mechanisms that respond to the progression of the optimization process. Among the evaluated techniques, the extended compact genetic algorithm integrated with simulated annealing demonstrates the highest level of accuracy, as shown in Table 1.

CONCLUSIONS

The research addresses the challenge of low efficiency in parametric identification methods used for approximating economic indicators within intelligent economic computer systems. To enhance the performance of numerical optimization techniques, three novel methods were developed, each integrating probabilistic modeling with simulated annealing principles.

The first method, grounded in Bayesian optimization, comprises four stages: reproduction, construction of a Bayesian network, generation of offspring based on this network, and a reduction phase to refine results. The second method employs a Markov-based optimization approach. It includes reproduction, formation of a Markov network, offspring generation using Gibbs sampling within the network structure, and a final reduction step. The third method builds on the extended compact genetic algorithm. It involves reproduction, modeling through the product of limiting distributions, generation of offspring guided by this model, and a reduction phase.

These optimization strategies are designed to explore the full search space during early iterations and to focus the search direction in later stages. This dual-phase approach enhances accuracy without requiring transformation of the fitness function.

Applied relevance of the proposed methods. The proposed optimization techniques broaden the applicability of metaheuristic frameworks grounded in probabilistic modeling, offering enhanced capabilities for intelligent economic computing systems. By improving search accuracy and adaptability, these methods contribute to the development of more efficient and scalable decision-support tools within complex socioeconomic environments.

Future research will focus on extending the applicability of the proposed approach to a wider spectrum of

DOI 10.15588/1607-3274-2025-4-8

artificial intelligence tasks, including pattern recognition, predictive modeling, and adaptive control systems. This opens promising avenues for interdisciplinary integration and real-world deployment across diverse domains.

REFERENCES

1. Talbi E.-G. Metaheuristics: from design to implementation. New Jersey : Wiley & Sons, 2009, 618 p.
2. Yu X. Introduction to evolutionary algorithms. London, Springer-Verlag, 2010, 433 p.
3. Engelbrecht A. Computational Intelligence: an introduction. West Sussex, Wiley & Sons, 2007, 630 p.
4. Subbotin S. Diagnostic Rule Mining Based on Artificial Immune System for a Case of Uneven Distribution of Classes in Sample, *Communications*, 2016, No. 3, pp. 11–13.
5. Nakib A. Metaheuristics for Medicine and Biology. Berlin, Springer-Verlag, 2017, 211 p.
6. Glover F., Kochenberger G. A. Handbook of Metaheuristics. Dordrecht, Kluwer Academic Publishers, 2003, 570 p.
7. Shvachych G., Ivaschenko O., Busygin V., Fedorov Ye. Parallel computational algorithms in thermal processes in metallurgy and mining, *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, 2018, No. 4, pp. 129–137.
8. Shlomchak G., Shvachych G., Moroz B., Fedorov E., Kozenkov D. Automated control of temperature regimes of alloyed steel products based on multi-processors computing systems, *Metalurgija,* 2019, Vol. 58, № 3–4, pp. 299–302.
9. Fedorov E., Lukashenko V., Patrushev V., Rudakov K., Mitsenko S. The method of intelligent image processing based on a three-channel purely convolutional neural network, *CEUR Workshop Proceedings*, 2018, Vol. 2255, pp. 336–351.
10. Yang X.-S. Nature-inspired Algorithms and Applied Optimization. Charm, Springer, 2018, 330 p.
11. Blum C. Hybrid Metaheuristics. Powerful Tools for Optimization. Charm, Springer, 2016, 57 p.
12. Martí R. Handbook of Heuristics. Charm, Springer, 2018, 1289 p.
13. Yang X.-S. Optimization Techniques and Applications with Examples. New Jersey, Wiley & Sons, 2018, 364 p.
14. Gendreau M. Handbook of Metaheuristics. New York, Springer, 2010, 640 p.
15. Bozorg-Haddad O. Meta-heuristic and Evolutionary Algorithms for Engineering Optimization. New Jersey, Wiley & Sons, 2017, 293 p.
16. Doerner K. F. Metaheuristics. Progress in Complex Systems Optimization. New York, Springer, 2007, 408 p.
17. de Castro L., von Zuben F. The Clonal Selection Algorithm with Engineering Applications, *Proceedings of the Genetic and Evolutionary Computation Conference "Workshop on Artificial Immune Systems and Their Applications*", 2000, pp. 36–39.
18. Chopard B., Tomassini M. An Introduction to Metaheuristics for Optimization. New York, Springer, 2018, 230 p.
19. Radosavljević J. Metaheuristic Optimization in Power Engineering. New York. The Institution of Engineering and Technology, 2018, 536 p.
20. White J., Garrett S. Improved Pattern Recognition with Artificial Clonal Selection, *Lecture Notes in Computer Science*, 2003, Vol. 2787, pp. 181–193.
21. de Castro L. N., von Zuben F. J. Learning and optimization using clonal selection principle, *IEEE Transactions on Evolutionary Computation,* 2002, Vol. 6, pp. 239–251.
22. Babayigit B. A clonal selection algorithm for array pattern nulling by controlling the positions of selected elements, *Progress In Electromagnetics Research,* 2008, Vol. 6, pp. 257–266.
23. Brownlee J. Clever Algorithms: Nature-Inspired Programming Recipes. Melbourne, Brownlee, 2011, 436 p.
24. Alba E., Nakib A., Siarry P. Metaheuristics for Dynamic Optimization. Berlin, Springer-Verlag, 2013, 398 p.
25. Pinto C. Wasp Swarm Algorithm for Dynamic MAX-SAT Problems, *Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms,* 2007, Part I, pp. 350–357.
26. Kumar S., Mahapatra R. P. Analytical Analysis of Two-Warehouse Inventory Model Using Particle Swarm Optimization, *Proceedings of Congress on Intelligent Systems*, 2021, Vol. 2, pp. 215–226
27. Du K.-L. Search and Optimization by Metaheuristics. Techniques and Algorithms Inspired by Nature. Charm, Springer, 2016, 434 p.
28. Sharma A., Sharma H., Khandelwal A., Sharma N. Designing Controller Parameter of Wind Turbine Emulator Using Artificial Bee Colony Algorithm, *Proceedings of Congress on Intelligent Systems*, 2020, pp. 143–151.
29. Nametala C., Faria W., Júnior B. On the performance of the Bayesian optimization algorithm with combined scenarios of search algorithms and scoring metrics, *Genetic Programming and Evolvable Machines*, 2022, No. 23, pp. 193–223.
30. Santana R., Bielza C., Larranaga P., Lozano J. A., Echegoyen C., Mendiburu A. Mateda-2.0: Estimation of Distribution Algorithms in MATLAB, *Journal of Statistica Software*, 2010, Vol. 35, pp. 1–30.
31. Nozari H., Aliahmadi A., Jafari-Eskandari M., Gh. Khaleghi An Extended Compact Genetic Algorithm for Milk Run Problem with Time Windows and Inventory Uncertainty, *International Journal of Applied Operational Research*, 2015. Vol. 5, № 2, pp. 35–48.

Received 10.06.2025.
Accepted 27.10.2025.

DOI 10.15588/1607-3274-2025-4-8

УДК 004.94

МЕТАЕВРИСТИЧНІ МЕТОДИ ПАРАМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ АПРОКСИМАЦІЙНОЇ МОДЕЛІ НА ОСНОВІ ЙМОВІРНИСНИХ МОДЕЛЕЙ

Григор О. О. – д-р політ. наук, професор, ректор Черкаського державного технологічного університету, Черкаси, Україна.

Федоров Є. Є. – д-р техн. наук, професор, професор кафедри статистики та прикладної математики Черкаського державного технологічного університету, Черкаси, Україна.

Лещенко М. М. – канд. екон. наук, доцент, доцент кафедри міжнародної економіки та бізнесу Черкаського державного технологічного університету, Черкаси, Україна.

Рудаков К. С. – канд. техн. наук, доцент, доцент кафедри робототехніки та спеціалізованих комп'ютерних систем, Черкаського державного технологічного університету, Черкаси, Україна.

Сахно Т. А. – PhD, доцент кафедри міжнародної економіки та бізнесу Черкаського державного технологічного університету, Черкаси, Україна.

АНОТАЦІЯ

Актуальність. Для покращення якості аналізу економічних індикаторів та підвищення ефективності методів числової оптимізації було запропоновано методи, засновані на синтезі алгоритмів імовірнісних моделей та моделюванні відпалу (байєсівська оптимізація з моделюванням відпалу, марковська оптимізація з моделюванням відпалу та розширений компактний генетичний алгоритм з моделюванням відпалу). Ці методи покращують точність пошуку та не потребують трансформації функції пристосованості завдяки принципу організації дослідження всього простору пошуку на початкових ітераціях та коригування напрямку пошуку на фінальних.

Метою роботи є підвищення ефективності параметричної ідентифікації моделі апроксимації економічних показників в інтелектуальних фінансових комп'ютерних системах за допомогою метаевристичних методів, заснованих на імовірнісних моделях та моделюванні відпалу.

Методи. У дослідженні застосовано метаевристичні методи оптимізації на основі імовірнісних моделей та моделювання відпалу для покращення ефективності параметричної ідентифікації моделі апроксимації економічних показників. Запропоновано три підходи: байєсівська оптимізація з моделюванням відпалу, марковська оптимізація з моделюванням відпалу та розширений компактний генетичний алгоритм з моделюванням відпалу. Ці методи підвищують точність пошуку, досліджуючи весь простір пошуку на початкових ітераціях та коригуючи напрямок на фінальних. Байєсівська оптимізація використовує байєсівську мережу для структурованого пошуку та вдосконалення рішень. Марковська оптимізація інтегрує квантування Гіббса у марковську мережу для покращення точності пошуку. Розширений компактний генетичний алгоритм застосовує моделі граничних розподілів для генерації оптимальних рішень. Запропоновані методи усувають необхідність трансформації функції пристосованості, оптимізуючи обчислювальну ефективність.

Результати. Запропоновані методи оптимізації покращили точність параметричної ідентифікації в інтелектуальних фінансових комп'ютерних системах. Поєднання імовірнісних моделей та моделювання відпалу підвищило ефективність пошуку без необхідності трансформації функції пристосованості.

Висновки. Запропоновані методи розширюють застосування метаевристик в економічному моделюванні, покращуючи обчислювальну ефективність. Перспективами подальших досліджень є використання запропонованих методів для більш широкого класу задач машинного навчання.

КЛЮЧОВІ СЛОВА: байєсівський алгоритм оптимізації, марковський алгоритм оптимізації, розширений компактний генетичний алгоритм, моделювання відпалу, числова оптимізація, параметрична ідентифікація моделі апроксимації економічних показників.

ЛІТЕРАТУРА

1. Talbi E.-G. Metaheuristics: from design to implementation / E.-G. Talbi. – New Jersey : Wiley & Sons, 2009. – 618 p.
2. Yu X. Introduction to evolutionary algorithms / X. Yu. – London : Springer-Verlag, 2010. – 433 p.
3. Engelbrecht A. Computational Intelligence: an introduction / A. Engelbrecht. – West Sussex : Wiley & Sons, 2007. – 630 p.
4. Subbotin S. Diagnostic Rule Mining Based on Artificial Immune System for a Case of Uneven Distribution of Classes in Sample / S. Subbotin // Communications. – 2016. – No. 3. – P. 11–13.
5. Nakib A. Metaheuristics for Medicine and Biology. / A. Nakib. – Berlin : Springer-Verlag, 2017. – 211 p.
6. Glover F. Handbook of Metaheuristics. / F. Glover, G. A. Kochenberger. – Dordrecht : Kluwer Academic Publishers, 2003. – 570 p.
7. Parallel computational algorithms in thermal processes in metallurgy and mining / [G. Shvachych, O. Ivaschenko, V. Busygin, Ye. Fedorov] // Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu. – 2018. – No. 4. – P. 129–137.
8. Automated control of temperature regimes of alloyed steel products based on multi-processors computing systems / [G. Shlomchak, G. Shvachych, B. Moroz et al.] // Metalurgija. – 2019. – Vol. 58, № 3–4. – P. 299–302.
9. The method of intelligent image processing based on a three-channel purely convolutional neural network / [E. Fedorov, V. Lukashenko, V. Patrushev et al.] //

DOI 10.15588/1607-3274-2025-4-8

CEUR Workshop Proceedings. – 2018. – Vol. 2255. – P. 336–351.

10. Yang X.-S. Nature-inspired Algorithms and Applied Optimization / X.-S. Yang. – Charm : Springer, 2018. – 330 p.
11. Blum C. Hybrid Metaheuristics. Powerful Tools for Optimization / C. Blum. – Charm : Springer, 2016. – 57 p.
12. Martí R. Handbook of Heuristics / R. Martí. – Charm : Springer, 2018. – 1289 p.
13. Yang X.-S. Optimization Techniques and Applications with Examples / X.-S. Yang. – New Jersey : Wiley & Sons, 2018. – 364 p.
14. Gendreau M. Handbook of Metaheuristics / M. Gendreau – New York : Springer, 2010. – 640 p.
15. Bozorg-Haddad O. Meta-heuristic and Evolutionary Algorithms for Engineering Optimization / O. Bozorg-Haddad. – New Jersey : Wiley & Sons, 2017. – 293 p.
16. Doerner K. F. Metaheuristics. Progress in Complex Systems Optimization / K. F. Doerner. – New York : Springer, 2007. – 408 p.
17. de Castro L. The Clonal Selection Algorithm with Engineering Applications / L. de Castro, F. von Zuben // Proceedings of the Genetic and Evolutionary Computation Conference “Workshop on Artificial Immune Systems and Their Applications”. – 2000. – P. 36–39.
18. Chopard B. An Introduction to Metaheuristics for Optimization / B. Chopard, M. Tomassini. – New York : Springer, 2018. – 230 p.
19. Radosavljević J. Metaheuristic Optimization in Power Engineering / J. Radosavljević. – New York : The Institution of Engineering and Technology, 2018. – 536 p.
20. White J. Improved Pattern Recognition with Artificial Clonal Selection / J. White, S. Garrett // Lecture Notes in Computer Science. – 2003. – Vol. 2787. – P. 181–193.
21. de Castro L. N. Learning and optimization using clonal selection principle / L. N. de Castro, F. J. von Zuben // IEEE Transactions on Evolutionary Computation. – 2002. – Vol. 6. – P. 239–251.
22. Babayigit B. A clonal selection algorithm for array pattern nulling by controlling the positions of selected elements / B. Babayigit // Progress In Electromagnetics Research. – 2008. – Vol. 6. – P. 257–266.
23. Brownlee J. Clever Algorithms: Nature-Inspired Programming Recipes / J. Brownlee. – Melbourne : Brownlee, 2011. – 436 p.
24. Alba E. Metaheuristics for Dynamic Optimization / E. Alba, A. Nakib, P. Siarry. – Berlin : Springer-Verlag. – 2013. – 398 p.
25. Pinto C. Wasp Swarm Algorithm for Dynamic MAX-SAT Problems / C. Pinto // Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms. – 2007. – Part I. – P. 350–357.
26. Kumar S. Analytical Analysis of Two-Warehouse Inventory Model Using Particle Swarm Optimization / S. Kumar, R. P. Mahapatra // Proceedings of Congress on Intelligent Systems. – 2021. – Vol. 2. – P. 215–226
27. Du K.-L. Search and Optimization by Metaheuristics. Techniques and Algorithms Inspired by Nature / K.-L. Du. – Charm : Springer, 2016. – 434 p.
28. Designing Controller Parameter of Wind Turbine Emulator Using Artificial Bee Colony Algorithm / [A. Sharma, H. Sharma, A. Khandelwal, N. Sharma] // Proceedings of Congress on Intelligent Systems. – 2020. – P. 143–151.
29. Nametala C. On the performance of the Bayesian optimization algorithm with combined scenarios of search algorithms and scoring metrics / C. Nametala, W. Faria, B. Júnior // Genetic Programming and Evolvable Machines. – 2022. – No. 23. – P. 193–223.
30. Mateda-2.0: Estimation of Distribution Algorithms in MATLAB / [R. Santana, C. Bielza, P. Larranaga et al.] // Journal of Statistica Software. – 2010. – Vol. 35. – P. 1–30.
31. An Extended Compact Genetic Algorithm for Milk Run Problem with Time Windows and Inventory Uncertainty / [H. Nozari, A. Aliahmadi, M. Jafari-Eskandari, Gh. Khaleghi] // International Journal of Applied Operational Research. – 2015. – Vol. 5, № 2. – P. 35–48.

DOI 10.15588/1607-3274-2025-4-8