Adaptive Metaheuristic Methods Based on the Covid-19 Virus' Behavior and Measures for Fight It

Eugene Fedorov[a], Olga Nechyporenko[a], Maryna Leshchenko[a], Tetyana Utkina[a], Kostiantyn Rudakov[a] and Nataliia Lada[a]

a Cherkasy State Technological University, Shevchenko blvd., 460, Cherkasy, 18006, Ukraine

Abstract

The research proposes the numerical optimization methods are based on metaheuristic methods on the basis of COVID-19 virus' behavior and measures to fight it. The novelty of research is defined by the fact, that to increase the numerical optimization efficiency, the following issues were developed. The anti-coronavirus optimization, which uses the normalized distance between people in an explicit form and the dynamic Levy flight parameter to modify the vector of human health characteristics at the stage of social distancing was improved. A power-law parameter to calculate the dynamic number of the weakest people was proposed. A local search based on the number account of days in quarantine to modify the vector of human health characteristics at the quarantine stage, arithmetic crossover based on the number of days in isolation to modify the vector of human health characteristics at the isolation stage were developed. A coronavirus optimization algorithm that uses dynamic mutation probability to modify the virion vector and coronavirus herd immunity optimizer that uses dynamic mutation probability to modify an individual's characteristic vector were improved. The proposed methods improve the speed and accuracy of finding a solution. The created metaheuristic methods based on the behavior of the COVID-19 virus and measures to fight it can be used in general and special-purpose intelligent systems.

Keywords 1

COVID-19, social distancing, infection mechanism, biological metaheuristics, numerical optimization

1. Introduction

The COVID-19 epidemic is no longer spreading around the world so quickly today, but it has already harmed the health and well-being of the population in different countries [1-4]. In 2023, there are over 700 million recovered cases, about 7 million deaths and over 200 thousand new cases of infection. The metaheuristics proposed in this research are based on COVID-19 control measures and the COVID-19 infection mechanism.

To reduce the probability of hitting a local extremum and more quickly find quasi-optimal solution metaheuristics (or modern heuristics) are used [5-6]. Metaheuristics are an extension of conventional heuristics and combine various heuristics [7-8]. Metaheuristics are used for continuous and discrete optimization [9-10].

2. Literature review

Modern metaheuristics have one or more disadvantages:

- only a general method's description is given without reference to the mathematical apparatus or the method's description cannot be used for several tasks [11-12];
- the finding solution process is not adaptive [13-14];
- the method will not necessarily converge [15-16];
- the method is intended only for binary potential solutions [17-18];
- the identifying parameter values process is difficult [19-20];
- only unconditional optimization problems can be solved [21-22];
- low accuracy of the method [23-24].

The task of creating effective metaheuristic optimization algorithms arises in this regard [25]. One of the popular metaheuristics are biological metaheuristics, of whom are viral ones, which allows solving numerical optimization problems [26-28].

The goal of the research is to increase the efficiency of numerical optimization through the use of metaheuristic methods based on the COVID-19 virus behavior and measures to fight it. To achieve this goal, it is necessary to solve the following tasks:
1. to create improved anti-coronavirus optimization;
2. to develop an improved algorithm for coronavirus optimization;
3. to create an improved coronavirus herd immunity optimizer;
4. to conduct a numerical study of the proposed viral methods.

3. Improved anti-coronavirus optimization

Basic anti-coronavirus optimization (anti-coronavirus optimization, ACVO) was suggested by Emami [4] and is based on measures to fight the COVID-19 virus spread. The basic algorithm has three main stages: social distancing, quarantine and isolation. During the social distancing phase, the algorithm tries to maintain a safe distance between people and limit close contacts. The algorithm quarantines suspicious people to prevent the spread of the disease during the quarantine stage. Moreover, some people who did not follow the medical protocols and contracted the virus must be taken care of for their full recovery. The algorithm takes care of infected people to restore their health in the isolation phase. The decision is made by an individual who is trying to stay healthy and slow the spread of COVID-19 by following containment protocols. The healthiest person is the best solution. The improved method uses the normalized distance between people explicitly to modify the vector of human health characteristics during the social distancing stage. Dynamic Levy flight parameter are used to modify the human health characteristics vector during the social distancing stage, i.e., Levy distribution is close to the long-tailed distribution at the early method stages (the search is more global) and the Levy distribution is close to the short-tail distribution the at the later stages of the method (the search is more local). There is a power-law parameter to calculate the dynamic number of the weakest people quarantined in the current iteration; local search includes the number of days in quarantine, to modify the human health characteristics vector at the quarantine stage; arithmetic crossing over includes the number of days in isolation, to modify the human health characteristics vector at the isolation stage.

3.1.   The algorithm for numerical functions optimization

1. Initialization

1.1. To set the minimum and maximum values of the Levy flight parameter $\lambda^{\min}$, $\lambda^{\max}$; the safe physical distance between people γ and basic reproductive number R_0 (it shows the average number of secondary infection cases caused by one primary infected person) are normalized; maximum duration of quarantine is E^Q; maximum duration of isolation is E^I; power-law parameter is η to calculate the number of the weakest people; a parameter δ uses to generate a new vector of human health characteristics, moreover $0 < \delta < 1$.

1.2. To set the maximum number of iterations N, population size K, length of the human health characteristics vector M, minimum and maximum values for the human health characteristics vector $x_j^{\min}, x_j^{\max}$, $j \in \overline{1, M}$.

1.3. To set the cost function (goal function)
$$F(x) \to \min_{x},$$
where x – is a human health characteristics vector.

1.4. To create the initial population P.

    1.4.1. Person number is $k = 1$, $P = \varnothing$.

    1.4.2. To generate the human health characteristics random vector x_k
$$x_k = (x_{k1}, ..., x_{kM}), \ x_{kj} = x_j^{\min} + (x_j^{\max} - x_j^{\min})U(0,1),$$
where $U(0,1)$ is a function returning a standard uniformly distributed random number.

    1.4.3. To initialize a person's health status (healthy)
$$s_k = 1.$$

    1.4.4. If $(x_k, s_k) \notin P$, then $P = P \bigcup \{(x_k, s_k)\}$, $k = k + 1$.

    1.4.5. If $k \leq K$, then go to step 1.4.2.

1.5. To create the quarantine list $Q = \varnothing$.

1.6. To create the isolation list $I = \varnothing$.

1.7. The Levy Flight Parameter initialization
$$\lambda = 0.$$

1.8. To determine the best person by goal function
$$k^* = \arg\min_{k} F(x_k), \ k \in \overline{1, K}, \ x^* = x_{k^*}.$$

2. Iteration number $n = 1$.

3. Social distancing.

    3.1. Person number $k = 1$.

    3.2. Person number $l = k + 1$.

    3.3. To calculate the normalized distance between a person k and a person l:
$$d_{kl} = \frac{\|x_k - x_l\|}{\| x^{\max} - x^{\min} \|}.$$

3.4. If $d_{kl} \geq \gamma$, then go to step 3.9.

3.5. The calculation of the control parameter for the local distance between a person k and a person l in his neighborhood based on Lévy flight (random walk); the Mantegna algorithm is used to calculate the step size Δ.
$$\Delta_{1j} = \delta d_{kl} \Delta_j, \ j \in \overline{1, M},$$
$$\Delta_j = \frac{\sigma_v N(0,1)}{\left|\sigma_u N(0,1)\right|^{1/\lambda(n)}},$$
$$\sigma_u = 1, \ \sigma_v = \left(\frac{\Gamma(1 + \lambda(n))}{\lambda\Gamma((1 + \lambda(n))/2)} \cdot \frac{\sin(\pi\lambda(n)/2)}{2^{(\lambda(n)-1)/2}}\right)^{1/\lambda(n)},$$
where $N(0,1)$ – a function returning a standard normally distributed number, $\Gamma(\cdot)$ – gamma function.

3.6. To calculate the normalized distance between a person k and the best person:
$$d_k = \frac{\|x^* - x_k\|}{\|x^{\max} - x^{\min}\|}.$$

3.7. The calculation of the global distance control parameter between a person k and the best person based on Lévy flight (random walk), the Mantegna algorithm is used to calculate the step size Δ is:
$$\Delta_{2j} = \delta d_k \Delta_j, \ j \in \overline{1, M},$$
$$\Delta_j = \frac{\sigma_v N(0,1)}{\left|\sigma_u N(0,1)\right|^{1/\lambda(n)}},$$

$$\sigma_u = 1, \; \sigma_v = \left(\frac{\Gamma(1+\lambda(n))}{\lambda\Gamma((1+\lambda(n))/2)} \cdot \frac{\sin(\pi\lambda(n)/2)}{2^{(\lambda(n)-1)/2}} \right)^{1/\lambda(n)}.$$

3.8. The modification of the human health characteristics vector:

   3.8.1. $\tilde{x}_{klj} = x_{kj} + \Delta_{1j} + \Delta_{2j}$, $j \in \overline{1,M}$.

   3.8.2. $\tilde{x}_{klj} = \max\{x_j^{\min}, \tilde{x}_{klj}\}$, $\tilde{x}_{klj} = \min\{x_j^{\max}, \tilde{x}_{klj}\}$, $j \in \overline{1,M}$.

3.9. Status modification (healthy):
$$s_k = 1.$$

3.10. if $l < K$, then $l = l+1$, go to step 3.3.

3.11. $l_k = \arg \min_{l \in \overline{k+1,K}} F(\tilde{x}_{kl})$, $\tilde{x}_k = \tilde{x}_{kl_k}$.

3.12. If $k < K$, then $k = k+1$, go to step 3.2.

4. Population reduction.

   4.1. To combine previous and new populations into one population
$$P = P \bigcup \tilde{P}.$$

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

   4.3. To keep in the population only K the first best people.

5. The calculation of the weakest people number:
$$K^Q(n) = \left(1 - \left(\frac{n}{N} \right)^\eta \right) R_0.$$

6. To add the weakest people to the quarantine list Q.

   6.1. $k = 1$, $m = |Q|$.

   6.2. Vector of human health characteristics before quarantine:
$$x_{0,m+k}^Q = \tilde{x}_{K-K^Q(n)+k}.$$

   6.3. Vector of human health characteristics during quarantine:
$$x_{m+k}^Q = \tilde{x}_{K-K^Q(n)+k}.$$

   6.4. A person's health status (in quarantine) initialization:
$$s_{m+k}^Q = 0.$$

   6.5. The initialization of the days number in quarantine:
$$e_{m+k}^Q = 0.$$

   6.6. $Q = Q \bigcup \{(x_{0,m+k}^Q, x_{m+k}^Q, s_{m+k}^Q, e_{m+k}^Q)\}$.

   6.7. If $k < K^Q(n)$, then $k = k+1$, go to step 6.2.

7. The population modification (the weakest people are quarantined):
$$\tilde{s}_k = 0, \; k \in \overline{K - K^Q(n)+1, K}.$$

8. Quarantine.

   8.1. Number of persons in quarantine is $k = 1$.

   8.2. The calculation of the human health modifiable characteristics number:
$$\tilde{M} = round(1 + (M-1)U(0,1)),$$

where $round()$ – function that rounds a number to the nearest integer.

   8.3. To generate a set of randomly selected human health characteristics numbers J.

      8.3.1. $i = 1$, $J = \varnothing$.

      8.3.2. $j = round(1 + (M-1)U(0,1))$.

      8.3.3. If $j \notin J$, then $J = J \bigcup \{j\}$, $i = i+1$.

      8.3.4. If $i \le \tilde{M}$, then go to step 8.3.2.

8.4. If $s_k^Q = 0 \wedge e_k^Q < E^Q$, then $x_{kj}^Q = x_{kj}^Q + \delta(x_j^{\max} - x_j^{\min})(2U(0,1)-1)\left(1 - \dfrac{e_k^Q}{E^Q}\right)$,

$x_{kj}^Q = \max\{x_j^{\min}, x_{kj}^Q\}$, $x_{kj}^Q = \min\{x_j^{\max}, x_{kj}^Q\}$, $j \in J$, $e_k^Q = e_k^Q + 1$, go to step 8.7.

8.5. If $F(x_k^Q) < F(x_{0k}^Q)$, to add a quarantined person to the population and removing from the quarantine list:

8.5.1. $x_{|P|+1} = x_k^Q$.

8.5.2. $s_{|P|+1} = 1$.

8.5.3. $P = P \bigcup \{(x_{|P|+1}, s_{|P|+1})\}$.

8.5.4. $Q = Q \setminus \{(x_{0k}^Q, x_k^Q, s_k^Q, e_k^Q)\}$.

8.6. If $F(x_k^Q) > F(x_{0k}^Q)$, to add a quarantined person to the isolation list and removing from the quarantine list:

8.6.1. Human health characteristics vector before isolation:
$$x_{0,|I|+1}^I = x_k^Q.$$

8.6.2. Human health characteristics vector during isolation:
$$x_{|I|+1}^I = x_k^Q.$$

8.6.3. A person's health status (in isolation) initialization:
$$s_{|I|+1}^I = -1.$$

8.6.4. The number of days in isolation initialization:
$$e_{|I|+1}^I = 0.$$

8.6.5. $I = I \bigcup \{(x_{|I|+1}^I, x_{|I|+1}^I, s_{|I|+1}^I, e_{|I|+1}^I)\}$.

8.6.6. $Q = Q \setminus \{(x_{0k}^Q, x_k^Q, s_k^Q, e_k^Q)\}$.

8.7. If $k < |Q|$, then $k = k + 1$, go to step 8.2.

9. Isolation.

9.1. Number of a person in isolation $k = 1$.

9.2. The calculation of the human health modifiable characteristics number:
$$\tilde{M} = round(1 + (M-1)U(0,1)).$$

9.3. To generate a set of randomly selected human health characteristics numbers J:

9.3.1. $i = 1$, $J = \varnothing$.

9.3.2. $j = round(1 + (M-1)U(0,1))$.

9.3.3. If $j \notin J$, then $J = J \bigcup \{j\}$, $i = i + 1$.

9.3.4. If $i \le \tilde{M}$, then go to step 9.3.2.

9.4. If $s_k^I = 0 \wedge e_k^I < E^I$, then $x_{kj}^I = x_{kj}^I + \delta(x_j^* - x_{kj}^I)U(0,1)\left(1 - \dfrac{e_k^I}{E^I}\right)$, $x_{kj}^I = \max\{x_j^{\min}, x_{kj}^I\}$,

$x_{kj}^I = \min\{x_j^{\max}, x_{kj}^I\}$, $j \in J$, $e_k^I = e_k^I + 1$, go to step 9.6.

9.5. If $F(x_k^I) < F(x_{0k}^I)$, to add to the population and removing from the isolation list a person who has undergone isolation:

9.5.1. $x_{|P|+1} = x_k^I$.

9.5.2. $s_{|P|+1} = 1$.

9.5.3. $P = P \bigcup \{(x_{|P|+1}, s_{|P|+1})\}$.

9.5.4. $I = I \setminus \{(x_{0k}^I, x_k^I, s_k^I, e_k^I)\}$.

9.6. If $k < |I|$, then $k = k + 1$, go to step 9.2.

10. Population reduction.

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

    10.2. To keep in the population only K the first best people.

11. The calculation of the Lévy flight parameter:

$$\lambda(n) = \lambda^{\min} + (\lambda^{\max} - \lambda^{\min})\left(\frac{n}{N}\right).$$

12. To determine the best person by goal function:

$$k^* = \arg\min_k F(x_k),\ k \in \overline{1,K},\ x^* = x_{k^*}.$$

13. To determine the global best vector of human health characteristics. If $F(x_{k^*}) < F(x^*)$, then

$x^* = x_{k^*}$.

    14. If $n < N$, then $n = n+1$, go to step 3.

The result is x^*.

4. The improved coronavirus optimization algorithm

The basic Coronavirus Optimization Algorithm (COVIDOA) was proposed by Khalid, Hosny, Mirjalili [5] and is based on the mechanism of human cell capture by coronavirus. The virus genome finds a ribosome in the human cell, which synthesizes many virus proteins based on the virus mRNA (matrix ribonucleic acid) of the, i.e., replicates mRNA. A viral mRNA is translated into viral proteins by reading the nucleotide sequence (building blocks) of the mRNA during the process of replication. The reading frame of the mRNA nucleotide sequence is shifted to the left/right by one. Each group of created virus proteins combines to form a virion (viral particle). The improved method uses dynamic mutation probability to modify the virion vector, i.e., the search is global at the early stages of the method, the search is local at the later stages of the method.

4.1. The algorithm for optimization of numerical functions

1. Initialization.

    1.1. To set the minimum and maximum values of the mutation probability $p^{\min}, p^{\max}$, β parameter, s shift parameter, δ parameter to generate a new human health characteristics vector, and $0 < \beta < 1$, $s \in \{-1,1\}$, $0 < \delta < 1$.

    1.2. Setting the maximum number of iterations N, K population size, M virion vector length, minimum and maximum values for the virion vector $x^{\min}, x^{\max}$.

    1.3. To set the cost function (goal function):

$$F(x) \to \min_x,$$

where x – is a virion vector.

    1.4. To create the initial population P.

        1.4.1. A virion number $k = 1$, $P = \varnothing$.

        1.4.2. Random virion vector x_k generation:

$$x_k = (x_{k1},...,x_{kM}),\ x_{kj} = x^{\min} + (x^{\max} - x^{\min})U(0,1),$$

where $U(0,1)$ – a function that returns a standard uniformly distributed random number.

        1.4.3. If $x_k \notin P$, then $P = P \bigcup \{x_k\}$, $k = k+1$.

        1.4.4. If $k \leq K$, then go to step 1.4.2.

    1.5. The mutation probability initialization:

$$p(n) = p^{\min}.$$

1.6. To determine the best virion according to its target function: $k^* = \arg\min\limits_{k} F(x_k)$, $k \in \overline{1, K}$, $x^* = x_{k^*}$.

2. Iteration number $n = 0$.

3. The calculation of the goal function average value:

$$F^{avg} = \frac{1}{K}\sum_{k=1}^{K} F(x_k).$$

4. The calculation of probabilities:

$$p_k = \exp\left(-\beta\frac{F(x_k)}{F^{avg}}\right), \quad k \in \overline{1, K}.$$

5. A virion number is $k = 1$.

6. To select of the parent virion number l based on a single application of roulette (proportional selection) and probabilities p_l, $l \in \overline{1, K}$.

7. Frameshift.

7.1. If $s = 1$, then $x_{k11} = x^{min} + (x^{max} - x^{min})U(0,1)$, $x_{k1j} = x_{k, j-1}$, $x_{k21} = x^{min} + (x^{max} - x^{min})U(0,1)$, $x_{k2j} = x_{k, j-1}$, $j \in \overline{2, M}$.

7.2. If $s = -1$, when $x_{k1M} = x^{min} + (x^{max} - x^{min})U(0,1)$, $x_{k1j} = x_{k, j+1}$, $x_{k2M} = x^{min} + (x^{max} - x^{min})U(0,1)$, $x_{k2j} = x_{k, j+1}$, $j \in \overline{1, M-1}$.

8. The crossing-over.

8.1. $\tilde{x}_{kj} = x_{k2j} + \delta(x_{k1j} - x_{k2j})(2U(0,1) - 1)$, $j \in \overline{1, M}$.

8.2. $\tilde{x}_{kj} = \max\{x_j^{min}, \tilde{x}_{kj}\}$, $\tilde{x}_{kj} = \min\{x_j^{max}, \tilde{x}_{kj}\}$, $j \in \overline{1, M}$.

9. Mutation.

9.1. $\lambda = U(0,1)$, $r_j = U(0,1)$, $\tilde{x}_{kj} = \begin{cases} r_j, & \lambda < p(n) \\ \tilde{x}_{kj}, & \lambda \geq p(n) \end{cases}$, $j \in \overline{1, M}$.

9.2. $\tilde{x}_{kj} = \max\{x_j^{min}, \tilde{x}_{kj}\}$, $\tilde{x}_{kj} = \min\{x_j^{max}, \tilde{x}_{kj}\}$, $j \in \overline{1, M}$.

10. If $k < K$, then $k = k + 1$, go to step 6.

11. Population reduction.

11.1. To combine previous and new populations into one population
$$P = P \bigcup \tilde{P}.$$

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

11.3. To leave only the K first best virions in the population.

12. The mutation probability calculation:

$$p(n) = p^{min} + (p^{max} - p^{min})\left(1 - \frac{n}{N}\right).$$

13. To determine the best virion according to its target function: $k^* = \arg\min\limits_{k} F(x_k)$, $k \in \overline{1, K}$, $x^* = x_{k^*}$.

14. To determine the global best virion vector. If $F(x_{k^*}) < F(x^*)$, then $x^* = x_{k^*}$.

15. If $n < N$, then $n = n + 1$, go to step 3.

The result is x^*.

5. The improved coronavirus herd immunity optimizer

The basic coronavirus herd immunity optimizer was proposed by Al-Betar, Alyasseri, Awadallah, Doush [6] and is based on the herd immunity concept and the social distancing concept as ways to

combat the coronavirus (COVID-19) pandemic. The rate of spread of coronavirus infection depends on how infected humans come into direct contact with other members of society. Herd immunity is a population condition in which the majority of the population is immune to infection, thereby preventing transmission of the disease. Three types of human individuals are used: susceptible, infected, and immunized to achieve herd immunity. The goal function evaluates immunity. The dynamic probability of mutation to modify the human characteristics vector is using in the improved method, i.e., the search is global in the early stages of the method, the search is local in the later stages of the method.

5.1. The numerical functions optimization algorithm

1. Initialization.

1.1. To set the minimum and maximum values of mutation probability $p^{\min}$, $p^{\max}$, the basic reproduction frequency α for consideration the spread of a virus pandemic between people, the maximum age of infected people β (when it is reached, individuals recover or die), and $0 < \alpha < 1$.

1.2. To set the maximum number of iterations N, population size K, length of the human characteristics vector M, minimum and maximum values for the human characteristics vector $x_j^{\min}, x_j^{\max}$, $j \in \overline{1, M}$.

1.3. To set the cost function (goal function):
$$F(x) \to \min_x,$$
where x – is a human characteristics vector.

1.4. To create the initial population P.

1.4.1. Person number $k = 1$, $P = \varnothing$.

1.4.2. Randomly creating a vector of human characteristics x_k
$$x_k = (x_{k1}, ..., x_{kM}), \ x_{kj} = x^{\min} + (x^{\max} - x^{\min})U(0,1),$$
where $U(0,1)$ – is a function that returns a standard uniformly distributed random number.

1.4.3. If $x_k \notin P$, then $P = P \bigcup \{x_k\}$, $k = k + 1$.

1.4.4. If $k \leq K$, then go to step 1.4.2.

1.5. State vector initialization:
$$s = (s_1, ..., s_K), \ s_k = 0.$$

1.6. Age vector initialization:
$$a = (a_1, ..., a_K), \ a_k = 0.$$

1.7. An infected person identification:
$$l = round(1 + (K-1)U(0,1)), \ s_l = 1,$$
where $round()$ – is the function that rounds a number to the nearest integer.

1.8. Initialization of mutation probability:
$$p(n) = p^{\min}.$$

1.9. To determine the best person by goal function:
$$k^* = \arg\min_k F(x_k), \ k \in \overline{1, K}, \ x^* = x_{k^*}.$$

2. Iteration number $n = 0$.

3. Person number $k = 1$.

4. The evolution of herd immunity.

4.1. $c_k = 0$.

4.2. $E = \{i \mid s_i = 1, i \in \overline{1, K}\}$.

4.3. $B = \{x_i \mid s_i = 0, i \in \overline{1, K}\}$.

4.4. $D = \{i \mid s_i = 2, i \in \overline{1, K}\}$, $v = \arg\min_{i \in D} F(x_i)$.

4.5. Person characteristics number $j = 1$.

4.6. $r = U(0,1)$.

4.7. If $r < \dfrac{\alpha}{3}$, then $l = round(1 + (|E| - 1)U(0,1))$, $\tilde{x}_{kj} = x_{kj} + (x_{kj} - x_{e_l j})(2U(0,1) - 1)$, $c_k = 1$.

4.8. If $\dfrac{\alpha}{3} \leq r < \dfrac{2\alpha}{3}$, then $m = round(1 + (|B| - 1)U(0,1))$, $\tilde{x}_{kj} = x_{kj} + (x_{kj} - x_{b_m j})(2U(0,1) - 1)$.

4.9. If $\dfrac{2\alpha}{3} \leq r < \alpha$, then $\tilde{x}_{kj} = x_{kj} + (x_{kj} - x_{vj})(2U(0,1) - 1)$.

4.10. If $r \geq \alpha$, then $\tilde{x}_{kj} = x_{kj}$.

4.11. $\tilde{x}_{kj} = \max\{x_j^{\min}, \tilde{x}_{kj}\}$, $\tilde{x}_{kj} = \min\{x_j^{\max}, \tilde{x}_{kj}\}$.

4.12. If $j < M$, then $j = j + 1$, go to step 4.6.

5. Mutation.

5.1. $\lambda = U(0,1)$, $r_j = U(0,1)$, $\tilde{x}_{kj} = \begin{cases} r_j, & \lambda < p(n) \\ \tilde{x}_{kj}, & \lambda \geq p(n) \end{cases}$, $j \in \overline{1, M}$.

5.2. $\tilde{x}_{kj} = \max\{x_j^{\min}, \tilde{x}_{kj}\}$, $\tilde{x}_{kj} = \min\{x_j^{\max}, \tilde{x}_{kj}\}$, $j \in \overline{1, M}$.

6. Modification of herd immunity population.

6.1. $F^{avg} = \dfrac{1}{K} \sum_{s=1}^{K} F(x_s)$.

6.2. If $F(\tilde{x}_k) \leq F(x_k)$, then $x_k = \tilde{x}_k$, otherwise $a_k = a_k + 1$.

6.3. If $F(\tilde{x}_k) < \dfrac{F(x_k)}{F^{avg}} \wedge s_k = 0 \wedge c_k = 1$, then $s_k = 1$, $a_k = 1$.

6.4. If $F(\tilde{x}_k) > \dfrac{F(x_k)}{F^{avg}} \wedge s_k = 1$, then $s_k = 2$, $a_k = 0$.

7. Critical conditions. If $(a_k \geq \beta) \wedge (s_k = 1)$, then $a_k = 0, s_k = 0$, $x_{kj} = x^{\min} + (x^{\max} - x^{\min})U(0,1)$, $j \in \overline{1, M}$.

8. If $k < K$, then $k = k + 1$, go to step 4.

9. To calculate the probability of mutation:

$$p(n) = p^{\min} + (p^{\max} - p^{\min})\left(1 - \frac{n}{N}\right).$$

10. To determine the best person by goal function:
$$k^* = \arg\min_k F(x_k), \ k \in \overline{1, K}, \ x^* = x_{k^*}.$$

11. To determine the global best vector of human characteristics. If $F(x_{k^*}) < F(x^*)$, then $x^* = x_{k^*}$

12. If $n < N$, then $n = n + 1$, go to step 3.

The result is x^*.

6. Numerical research

Numerical research of the proposed metaheuristic methods was carried out using the Python package in the Google Colab environment.

The minimum and maximum value of the Levy flight parameter is $\lambda^{\min} = 1$, $\lambda^{\max} = 2$, safe normalized physical distance between people is $\gamma = 0.5$, power-law parameter for the dynamic number of the weakest people calculating is $\eta = 1.5$, basic reproductive number is $R_0 = 2.5$, maximum duration of quarantine is $E^Q = 5$, maximum duration of isolation is $E^I = 10$, a parameter for generating a new human health characteristics vector is $\delta = 0.1$. All of them were used for anti-coronavirus optimization improving.

The minimum and maximum probability value is $p^{\min} = 0.1$, $p^{\max} = 0.9$, parameter is $\beta = 0.5$, shift parameter is $s = 1$, parameter for generating a new human health characteristics vector is $\delta = 0.1$. All of them were used for coronavirus optimization algorithm improving.

The minimum and maximum probability value is $p^{\min} = 0.1$, $p^{\max} = 0.9$, basic reproduction rate is $\alpha = 0.05$, maximum age of infected people is $\beta = 100$. All of them was used for improving of the coronavirus herd immunity optimizer.

The population size is $K = 100$, the length of the solution vector is $M = 10$ for all three methods.

The Lévy flight parameter increasing function is determined by the formula: $\lambda(n) = \lambda^{\min} + (\lambda^{\max} - \lambda^{\min})\left(\dfrac{n}{N}\right)$ and it's linear.

The Levy flight parameter increases with the iteration number increase according to the dependence of the increase in the Levy flight parameter on the iteration number (Fig. 1).

The mutation probability decreasing function is determined by the formula: $p(n) = p^{\min} + (p^{\max} - p^{\min})\left(1 - \dfrac{n}{N}\right)$ and it's linear.

The probability decreases with increasing iteration number according to the dependence of the mutation probability decrease on the iteration number (Fig. 2).

The solution search was carried out using the Rasstrigin test function $F(x) = 10M + \sum_{j=1}^{M}(x_j^2 - 10\cos(2\pi x_j))$ with a global minimum $x = \mathbf{0}$ for all three methods.

The results of comparing between the proposed and traditional methods (which are based on measures to combat COVID-19 and the mechanism COVID-19 infection) are presented in Tables 1-3.

Table 1

The comparison between the proposed and traditional anti-coronavirus optimization

The iterations number		Standard derivation	
For the proposed method	For the current method	For the proposed method	For the current method
1000	2000	0.02	0.07

Table 2

The comparison between the proposed and traditional coronavirus optimization algorithm

The iterations number		Standard derivation	
For the proposed method	For the current method	For the proposed method	For the current method
500	1000	0.06	0.11

Table 3

The comparison between the proposed and traditional coronavirus herd immunity optimizer

The iterations number		Standard derivation	
For the proposed method	For the current method	For the proposed method	For the current method
750	1500	0.04	0.09

7. Discussion of results

1. The anti-coronavirus optimization method is more accurate and faster compared to the existing one due to:
- the normalized distance between people explicitly for modifying the human health characteristics vector at the stage of social distancing (Table 1);
- the dynamic Levy flight parameter to modify the human health characteristics vector at the stage of social distancing, i.e., the Levy distribution is close to the long-tailed distribution

at the early stages of the method (the search is more global), the Levy distribution is close to the short-tail distribution at the later stages of the method (the search is more local) (Table 1);

- the power-law parameter for the calculating of the weakest people dynamic number quarantined at the current iteration (Table 1);
- the local search, that considers the number of days in quarantine, to modify the human health characteristics vector at the quarantine stage (Table 1);
- the arithmetic crossing over, that considers the number of days in isolation, to modify the human health characteristics vector at the isolation stage (Table 1).

2. The coronavirus optimization algorithm makes it possible to use: dynamic mutation probability for modifying the virion vector, i.e., the search is global at the early method stages and the search is local at the later method stages. It makes the proposed method more accurate and faster compared to the existing one (Table 2).

3. The coronavirus herd immunity optimizer allows to use: dynamic mutation probability to modify the human characteristics vector, i.e., the search is global at the early method stages and the search is local at the later method stages; it makes the proposed method more accurate and faster compared to the existing one (Table 3).

8. Acknowledgements

9. References

[1] C. Sheridan, "Coronavirus and the race to distribute reliable diagnostics," Nature Biotechnology, vol. 38, no. 4, p. 382, 2020.

[2] G. Lippi, A.-M. Simundic, and M. Plebani, Potential preanalyticaland analytical vulnerabilities in the laboratory diagnosis of coronavirus disease 2019 (COVID-19) / Clinical Chemistry and Laboratory Medicine (CCLM), vol. 58, no. 7, p. 1, 2020.

[3] G. Lippi and M. Plebani, A six-sigma approach for comparing diagnostic errors in healthcare— where does laboratory medicine stand? / Annals of Translational Medicine, vol. 6, no. 10, 2018.

[4] B. A. Oliveira, L. C. Oliveira, E. C. Sabino, and T. S. Okay, SARS-CoV-2 and the COVID-19 disease: a mini review on diagnostic methods / Revista Do Instituto de Medicina Tropical de Sao Paulo, vol. 62, 2020.

[5] X.-S. Yang, Nature-inspired Algorithms and Applied Optimization / Charm : Springer, p. 330, 2018. doi:10.1007/978-3-642-29694-9

[6] A. Nakib, El-G. Talbi, Metaheuristics for Medicine and Biology / Berlin : Springer-Verlag, p. 211, 2017. doi:10.1007/978-3-662-54428-0

[7] S. Subbotin, A. Oliinyk, V. Levashenko, E. Zaitseva, Diagnostic Rule Mining Based on Artificial Immune System for a Case of Uneven Distribution of Classes in Sample / Communications, vol. 3. P. 3–11 2016.

[8] X.-S. Yang, Optimization Techniques and Applications with Examples / Hoboken, New Jersey : Wiley & Sons, p. 364, 2018. DOI: 10.1002/9781119490616

[9] T. Neskorodieva, E. Fedorov, Method for automatic analysis of compliance of settlements with suppliers and settlements with customers by neural network model of forecast / Advances in Intelligent Systems and Computing, 1265 AISC, pp. 156–165, 2021.

[10] A. Vasuki, Nature Inspired Optimization Algorithms / Boca Raton, FL: Chapman and Hall/CRC, p. 260, 2020.

[11] H. Wang, M. Huang, J. Wang, An effective metaheuristic algorithm for flowshop scheduling with deteriorating jobs / Journal of Intelligent Manufacturing, vol. 30, P. 2733–2742, 2019. doi:10.1007/s10845-018-1425-8

[12] C. Blum, G. R. Raidl, Hybrid Metaheuristics. Powerful Tools for Optimization / Charm : Springer, p. 157, 2016. doi:10.1007/978-3-319-30883-8

[13] R. Martí, P. M. Pardalos, M. G. C. Resende, Handbook of Heuristics / Charm : Springer, p. 1289, 2018. doi:10.1007/978-3-319-07124-4

[14] B. Chopard, M. Tomassini, An Introduction to Metaheuristics for Optimization / New York : Springer, p. 230, 2018. doi:10.1007/978-3-319-93073-2

[15] J. Radosavljević, Metaheuristic Optimization in Power Engineering / – New York : Institution of Engineering and Technology, p. 536, 2018. doi:10.1049/PBPO131E

[16] N.S. Jaddi, J. Alvankarian, S. Abdullah, Kidney-inspired algorithm for optimization problems / Communications in Nonlinear Science and Numerical Simulation, vol. 42, P. 358–369, 2017.

[17] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization / IEEE Computational Intelligence Magazine, vol. 12, no. 12, P. 73–87, 2017

[18] E. Fedorov, O. Nechyporenko, Methods for Solving the Traveling Salesman Problem Based on Reinforcement Learning and Metaheuristics / CEUR Workshop Proceedings, vol. 2022, P. 94–103, 2022. URL: http s://ceur-ws.org/Vol-3309/paper8.pdf

[19] T. Neskorodieva, E. Fedorov, M. Chychuzhko, V. Chychuzhko, Metaheuristic Method for Searching Quasi-Optimal Route Based On the Ant Algorithm and Annealing Simulation // Radioelectronic and Computer Systems, p. 92–102, 2022.

[20] A. Shukla, R. Tiwari, Discrete Problems in Nature Inspired Algorithms / Boca Raton, FL: Chapman and Hall/CRC, p. 310, 2019.

[21] A. Slowik, Swarm Intelligence Algorithms A Tutorial / Boca Raton, FL: Chapman and Hall/CRC, p. 362, 2021.

[22] O. Bozorg Haddad, M. Solgi, H. Loaiciga, Meta-heuristic and Evolutionary Algorithms for Engineering Optimization / Hoboken, New Jersey : Wiley & Sons, p. 293, 2017. DOI: 10.1002/9781119387053

[23] K.-L. Du, M. S. Swamy, Search and Optimization by Metaheuristics. Techniques and Algorithms Inspired by Nature / Charm : Springer, p. 434, 2016. doi:10.1007/978-3-319-41192-7

[24] A. Kaveh, T. Bakhshpoori, Metaheuristics Outlines, MATLAB Codes and Examples / Cham: Springer, p. 190, 2019.

[25] A. Nayyar, D.-N. Le, N.G. Nguyen, Advances in Swarm Intelligence for Optimizing Problems in Computer Science / Boca Raton, FL: Chapman and Hall/CRC, p. 298, 2019.

[26] H. Emami, Anti-coronavirus optimization algorithm / Soft Computing, vol. 26, P. 4991–5023, 2022.

[27] A.M. Khalid, K.M. Hosny, S. Mirjalili, COVIDOA: a novel evolutionary optimization algorithm based on coronavirus disease replication lifecycle / Neural Computing and Applications, vol. 34, P. 22465–22492, 2022.

[28] A. Al-Betar, Z.A.A. Alyasseri, M.A. Awadallah, I.A. Doush, Coronavirus herd immunity optimizer (CHIO) / Neural Computing and Applications, vol. 33, P. 5011–5042, 2021.