Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 18(7), 264–276
https://doi.org/10.12913/22998624/193195
ISSN 2299-8624, License CC-BY 4.0

Received: 2024.08.22
Accepted: 2024.09.23
Published: 2024.10.06

Optimization of Quality Control Processes Using the NPGA Genetic Algorithm

Andrzej Chmielowiec[1*], Sylwia Sikorska-Czupryna[1], Leszek Klich[1], Weronika Woś[1], Paweł Kuraś[2]

[1] Faculty of Mechanics and Technology, Rzeszow University of Technology, ul. Kawiatkowskiego 4, 37-450 Stalowa Wola, Poland

[2] Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, ul. Wincentego Pola 2, 35-959 Rzeszow, Poland

* Corresponding author's e-mail: achmie@prz.edu.pl

ABSTRACT

In the article, the problem of multi-criteria optimization of quality control mechanisms is analyzed. The presented method assumes the use of the NPGA genetic algorithm to simultaneously manage costs and the level of detecting non-conformities. The main assumption of the presented approach is to treat individual quality control procedures as vectors, whose elements are probability generating functions of defect detection. Each of these procedures generates certain operational costs and covers specific types of defects within its scope. The task of the presented algorithm is to indicate which procedure and to what extent should operate to ensure an appropriate level of non-conformity detection while minimizing costs. The article presents the theoretical foundations of the developed algorithm and the results of its implementation. The software has been developed in C++ with a particular focus on performance aspects. Its essence lies in the implementation of data structures introduced in the theoretical part, as well as methods for their rapid processing. Thanks to this approach, the entire program is scalable and can be used to solve multidimensional optimization problems. The presented approach may also find application in other areas of enterprise management. This will be possible primarily in cases where the effectiveness of procedures or devices is primarily evaluated based on probability. Therefore, the presented methods can provide effective optimization of other areas related to enterprise management.

Keywords: quality control, quality management, enterprise management, multi-criteria process optimization, genetic algorithm, NPGA.

INTRODUCTION

Quality control systems play a key role in ensuring high quality products and services. However, the implementation and maintenance of such systems often involves significant costs, which include hardware, software, human resources and other operational expenses [1–5]. Optimizing these costs, while ensuring system efficiency and effectiveness, is a significant challenge for companies and organizations. In addition to direct cost minimization, improving the overall efficiency of quality control systems is crucial. To achieve this, multi-criteria optimization methods should be used. For example, Nikolay et al. [6] examined the effectiveness of multi-criteria optimization methods to increase the efficiency of power system operating states, and developed an approach to comparing the efficiency of multi-criteria optimization algorithms. Al-Zuheri [7] utilized genetic algorithms to optimize the supply chain while considering disruption risks. On the other hand, Iwankowicz and Sekulski [8] presented the application of genetic algorithms to solve the traveling salesman problem, which is an NP-hard and can serve as a basis for solving entirely different problems. Genetic algorithms have also been applied in the optimization of sample size for control charts (SPC) in quality control processes [9] and in the optimization of quality control procedures [10].

In the article [11] an attempt was made to create a multi-objective optimization model that can take into account various aspects such as profit, time, and others using genetic algorithms and objective programming. However, the developed model first optimizes each criterion independently and then determines the optimal solution by taking into account all aspects together. It turns out that multi-objective optimization problems are often considered using a genetic algorithm. More precisely, the evolutionary NSGA-II algorithm. This topic is discussed, for example, by Hinojosa et al. [12]. However the authors of this paper have not found a publication using the Niched Pareto Genetic Algorithm to optimize quality control processes.

Therefore, the objective of this research is to apply the Niched Pareto Genetic Algorithm (NPGA) to optimize quality control processes, addressing both cost reduction and efficiency improvements. While genetic algorithms have been widely used in quality control system optimization, the specific application of NPGA has not been explored in this context. The uniqueness of NPGA lies in its ability to maintain population diversity through niche pressure, making it particularly suitable for multi-objective optimization problems, such as balancing defect detection rates and operational costs. Genetic algorithms, inspired by evolutionary processes in nature, offer a promising approach to solving this problem. They are optimization techniques that use mechanisms such as mutation, crossover and selection to find optimal or approximate solutions to complex optimization problems. Following John Holland's pioneering theoretical work, the last decade has seen an extensive literature devoted to their application to real-world problems. The basics of the method can be found in [13]. Meanwhile, some applications, in various contexts, are included in ref. [14]. In the context of quality control systems, genetic algorithms can be used to optimize various factors, such as the number of inspection stations, inspection frequency, resource allocation or maintenance schedules. The scientific literature provides many examples of the successful application of genetic algorithms in optimizing the cost of quality control systems in various industries. Zhou et al. [15] proposed a model based on a genetic algorithm for cost optimization in a quality control system for semiconductor manufacturing. Experimental results showed significant cost savings compared to traditional methods. In contrast, Duffuaa et al. [16] developed a quality control strategy using genetic algorithms for manufacturing systems with multiple stages, which reduced total inspection costs.

In addition to direct cost optimization, genetic algorithms can also be used to improve the overall performance of quality control systems. Chen and Tsai [17] presented a model based on a genetic algorithm for optimizing inspection scheduling in quality control systems with time and resource constraints. The results showed increased efficiency and reduced costs compared to traditional scheduling methods. Combining genetic algorithms with other optimization techniques or artificial intelligence methods is also an important aspect. For example, Vanini et al. [18] proposed a hybrid model combining a genetic algorithm with a neural network to optimize maintenance schedules in a quality control system, which resulted in reduced costs and increased system reliability.

This paper fills a gap in the literature by providing a novel approach to multi-criteria optimization in quality control systems using NPGA. Unlike traditional genetic algorithms, which may struggle with maintaining diversity in high-dimensional optimization problems, NPGA offers an improved solution by focusing on Pareto dominance and diversity preservation. This ensures that the optimization process does not prematurely converge on suboptimal solutions, a common issue in other approaches.

Despite its many benefits, the application of genetic algorithms in cost optimization of quality control systems also comes with some challenges. Proper definition of the objective function and constraints of the optimization problem is crucial to obtain satisfactory results [19]. In addition, the proper selection of genetic algorithm parameters, such as population size, mutation and crossover probabilities, can significantly affect the efficiency and effectiveness of the optimization process [20].

MATERIALS AND METHODS

Imagine a situation in which a certain defect goes through the same quality control procedure multiple times. It can be assumed that no single procedure generally guarantees the complete detection of a defect at each verification. Thus, for each type of defect, we obtain a certain distribution that determines the probability of multiple detection of a defect under a given procedure. We can write this distribution in the form of a probability formation function, which we will denote by $\mathcal{C}$.

Extended representation of the $\mathcal{C}$ function

From a mathematical and computational point of view, the function $\mathcal{C}$ can be represented as a vector of power series. We will denote such an extended form of the function $\mathcal{C}$ by $\mathcal{C}_{\mathcal{E}}$ and define it as:

$$\mathcal{C}_{\mathcal{E}}(T) = (t_1, t_2, \dots, t_n) \tag{1}$$

where: t_i are power series of the form

$$t_i = t_{i,0} + t_{i,1}X + t_{i,2}X^2 + \cdots, \tag{2}$$

for which the condition is met

$$\textstyle\sum_{j=0}^{\infty} t_{i,j} = 1 \tag{3}$$

The relationship between $\mathcal{C}$ and $\mathcal{C}_{\mathcal{E}}$ is as follows

$$\mathcal{C}(T) = \frac{w_1t_1+w_2t_2+\cdots+w_nt_n}{w_1+w_2+\cdots+w_n} = \textstyle\sum_{j\,=\,0}^{\infty} p_j\, X^j \tag{4}$$

where: $w_1, w_2, \dots, w_n$ are the weights of the individual coordinates. Thus, it can be said that the function $\mathcal{C}$ is a weighted average of its extended representation. Having access to such a representation of $\mathcal{C}$, we can define the values of the function $\mathcal{C}_{k+}$

$$\mathcal{C}_{k+}(T) = 1 - \textstyle\sum_{j=0}^{k-1} p_j \tag{5}$$

which define the probability of k-times a defect is detected by a given procedure. Thus, in particular, we have $\mathcal{C}_{1+}(T) = 1 - p_0$.

This representation is convenient from a computational point of view, as it provides a simple way to determine the $\mathcal{C}$ function for a larger number of procedures. Indeed, if we have given two quality control procedures *R* i *S*, for which:

$$\mathcal{C}_{\mathcal{E}}(R) = (r_1, r_2, \dots, r_n) \tag{6}$$

$$\mathcal{C}_{\mathcal{E}}(S) = (s_1, s_2, \dots, s_n) \tag{7}$$

it is:

$$\mathcal{C}_{\mathcal{E}}(RS) = \mathcal{C}_{\mathcal{E}}(R) + \mathcal{C}_{\mathcal{E}}(S) = (r_1s_1, r_2s_2, \dots, r_ns_n) \tag{8}$$

Thus, as can be seen in this form, it is quite easy to combine different quality control procedures. This approach also allows us to introduce into the space of procedures something that has some of the characteristics of a linear space. Indeed, we have the following properties:

1. The connectivity of the operation

$$\mathcal{C}_{\mathcal{E}}(R) + \big(\mathcal{C}_{\mathcal{E}}(S) + \mathcal{C}_{\mathcal{E}}(T)\big) = \big(\mathcal{C}_{\mathcal{E}}(R) + \mathcal{C}_{\mathcal{E}}(S)\big) + \mathcal{C}_{\mathcal{E}}(T) = \mathcal{C}_{\mathcal{E}}(RST) \tag{9}$$

is due to the connectivity of the multiplication of formal series.

2. Alternation of operation

$$\mathcal{C}_{\mathcal{E}}(R) + \mathcal{C}_{\mathcal{E}}(S) = \mathcal{C}_{\mathcal{E}}(S) + \mathcal{C}_{\mathcal{E}}(R) \tag{10}$$

is due to the alternation of multiplication of formal series.

3. Existence of an identity element

$$\mathcal{C}_{\mathcal{E}}(I) = (1{,}1, \dots{,}1) \tag{11}$$

4. Multiplication by a scalar is separable from addition of vectors

$$\alpha \cdot \big(\mathcal{C}_{\mathcal{E}}(R) + \mathcal{C}_{\mathcal{E}}(S)\big) = ((r_1s_1)^{\alpha}, (r_2s_2)^{\alpha}, \dots, (r_ns_n)^{\alpha}) \tag{12}$$

$$= (r_1^{\alpha}, r_2^{\alpha}, \dots, r_n^{\alpha})(s_1^{\alpha}, s_2^{\alpha}, \dots, s_n^{\alpha}) \tag{13}$$

$$= \alpha \cdot \mathcal{C}_{\mathcal{E}}(R) + \alpha \cdot \mathcal{C}_{\mathcal{E}}(S) \tag{14}$$

5. Multiplication by a vector is separable from addition of scalars

$$(\alpha+\beta)\cdot\mathcal{C}_{\mathcal{E}}(R)=\left(r_1^{\alpha+\beta},r_2^{\alpha+\beta},\dots,r_n^{\alpha+\beta}\right) \tag{15}$$

$$=(r_1^{\alpha},r_2^{\alpha},\dots,r_n^{\alpha})\left(r_1^{\beta},r_2^{\beta},\dots,r_n^{\beta}\right) \tag{16}$$

$$=\alpha\cdot\mathcal{C}_{\mathcal{E}}(R)+\beta\cdot\mathcal{C}_{\mathcal{E}}(R) \tag{17}$$

6. Multiplication by a scalar is consistent with multiplication of scalars

$$\alpha\cdot\left(\beta\cdot\mathcal{C}_{\mathcal{E}}(R)\right)=\left(r_1^{\alpha\beta},r_2^{\alpha\beta},\dots,r_n^{\alpha\beta}\right) \tag{18}$$

$$=(\alpha\beta)\cdot\mathcal{C}_{\mathcal{E}}(R) \tag{19}$$

7. Multiplication by a scalar has an identity element

$$1\cdot\mathcal{C}_{\mathcal{E}}(R)=\mathcal{C}_{\mathcal{E}}(R) \tag{20}$$

Unfortunately, the opposite vectors are lacking in the presented space, as they do not have a proper interpretation in quality control terms. In addition, scalars should be limited to natural numbers, since they actually correspond to the number of uses of a given quality control procedure.

Nevertheless, even with this limited algebraic structure, we can try to adapt some methods of linear algebra to the process of optimizing the distribution of functions $\mathcal{C}$.

Multicriteria optimization

Genetic algorithms use the mechanisms of natural evolution: selection, survival of the best adapted individuals, reproduction. The essence of the genetic algorithm is that the space is not searched directly. A small population is selected at random. Searching is done through the mechanisms of evolution and natural selection. In each iteration of the genetic algorithm, the processed population of solutions becomes a population of increasingly well-adapted individuals, representing solutions closer and closer to the optimum. Thus, genetic algorithms are optimization methods aimed at finding the global optimum of a set of real objective functions of one or more decision variables, which may be subject to various linear or nonlinear constraints.

In multi-criteria optimization problems (MOP), conflicts between objectives usually prevent having a single optimal solution, but rather a set of compromise solutions, called the Pareto optimal [21]. A Pareto solution is said to be dominant over another solution if it is not inferior in any objective and is strictly superior in at least one objective. Solutions in the Pareto set are not dominated by any other solution of the admissible solution space [22, 23].

In many MOPs, it is difficult to obtain a complete and accurate set of Pareto set solutions, and thus it is reasonable to obtain an approximation to it. To this end, multi-criteria evolutionary algorithms (MOEAs) have been shown to be well suited to complex MOPs with two or three [22] objectives. These algorithms handle multi-criteria problems by simulating the basic principles of the evolutionary process on a set of individuals (solutions), i.e., an evolutionary population, using so-called evolutionary operators (assignment, selection, crossover, mutation and elitism) [22, 23]. In general, MOEAs vary in their method of assignment, but most of them belong to a family called "Pareto-based" which uses the concept of Pareto dominance as the basis for distinguishing between solutions to guide the search for them [22].

Recently, several researchers have pointed out the difficulty of convergence of MOEAs when they are used to solve multi-object problems, i.e., problems having 4 or more objective functions [24–30]. In the case of Pareto-based MOEAs, these difficulties are inherent in the fact that as the number of targets increases, the proportion of non-dominant elements in the population increases, making it increasingly difficult to distinguish between solutions using only a dominance relationship [23, 25].

In addition, several MOEAs are based on data structures and subroutines with complexity that grows exponentially in the number of objectives [24]. Moreover, the number of solutions needed to approximate the entire Pareto front (set) grows exponentially with the dimensionality of the objective space [31, 32].

Once the MOEA obtains a Pareto set approximation, it is assumed that the decision-maker chooses one solution. At this stage, visualization of solution alternatives becomes very important. Although some methods have been proposed for this purpose [33– 37], a simple, intuitive way to represent solutions in objective space with four or more objectives is still lacking.

The success of a search with a MOGA algorithm depends largely on the ability of the algorithm to maintain genetic diversity in the population [38]. In this regard, standard MOGA algorithms based on the Pareto set [13] may encounter difficulties in maintaining genetic diversity when searching a multidimensional feature space [39], so that solutions found at convergence may not uniformly represent the Pareto set, and will include only a portion of it [39, 40].

To solve this problem, the paper adopts the Niched Pareto MOGA [40] to exploit its ability to evolve populations toward alternative, equivalent solutions of subsets of features that yield a well-distributed, representative description of the Pareto set of non-dominated solutions. This is achieved by applying "niche pressure" at the parental selection stage, so that individuals with less crowded neighborhoods in the space of the objective function are preferentially selected as parents, thus producing more offspring in subsequent generations: this results in a more even distribution of the population in the space of the objective function [40].

NPGA algorithm for optimizing quality control procedures

In order to use the NPGA algorithm to optimize a quality control system, it is necessary to write it in the language of $\mathcal{C}$ functions. Therefore, it is necessary to define what a quality control system is from the point of view of a genetic optimizer.

A quality control system is a sequence of coefficients $Q = (k_1, k_2, k_3, \dots, k_i)$, where $k_i$are non-negative integers denoting the number of uses of a given control procedure R_i. The number $k_i$is called the multiplicity of the procedure or the multiplicity of the gene R_i.

The function $\mathcal{C}$ of the quality control system $Q = (k_1, k_2, k_3, \dots, k_i)$ is calculated on the basis of the extended representation $\mathcal{C}$, which is of the form

$$\mathcal{C}_{\mathcal{E}}(Q) = k_1 \cdot \mathcal{C}_{\mathcal{E}}(R_1) + k_2 \cdot \mathcal{C}_{\mathcal{E}}(R_2) + \cdots + k_i \cdot \mathcal{C}_{\mathcal{E}}(R_i) \qquad (21)$$

The relationship between $\mathcal{C}(Q)$, a $\mathcal{C}_{\mathcal{E}}(Q)$ is determined by the equation (4). Therefore, a quality control system can be treated as a new procedure – a procedure that is a combination of many other procedures (including their multiples).

Quality control system matching is a function designed to determine how a given control system fits into the user's expectations. This function is based on a sequence of user-specified weights $(\mu_1, \mu_2, \dots, \mu_i) \in [0,1]^i$ (the weights are independent for each optimization process), and its value is determined by the formula

$$u(Q) = \frac{\mu_1 k_1 + \mu_2 k_2 + \cdots + \mu_i k_i}{k_1 + k_2 + \cdots + k_i} \cdot \mathcal{C}_{1+}(Q) \qquad (22)$$

The cost of a quality control system is defined by the formula

$$c(Q) = k_1 C_1 + k_2 C_2 + \cdots + k_i C_i \qquad (23)$$

where: C_i denotes the cost of implementing a single control procedure.

A random quality control system is such a sequence $(k_1, k_2, \dots, k_i)$ for which $k_i = 7 - \lfloor \log_2(r_i) \rfloor$ and r_i is a random natural number from the interval [0.255]. From such a definition, it follows that random quality control systems have coefficients k_i belonging to the set {0,1,...,7}. Therefore, the probability that the tuple of a procedure/gene k_i has a certain value is as follows:

- $\mathcal{P}(k_i = 0) = \frac{1}{2}$,
- $\mathcal{P}(k_i = 1) = \frac{1}{4}$,
- $\mathcal{P}(k_i = 2) = \frac{1}{8}$,
- $\mathcal{P}(k_i = 3) = \frac{1}{16}$,
- $\mathcal{P}(k_i = 4) = \frac{1}{32}$,
- $\mathcal{P}(k_i = 5) = \frac{1}{64}$,
- $\mathcal{P}(k_i = 6) = \frac{1}{128}$,
- $\mathcal{P}(k_i = 7) = \frac{1}{128}$.

Recombination of quality control systems is otherwise known as system crossover. As a result of this operation from two systems $Q_r = (r_1, r_2, \dots, r_i)$ i $Q_s = (s_1, s_2, \dots, s_i)$ we create two descendant systems $Q_x = (x_1, x_2, \dots, x_i)$ and $Q_y = (y_1, y_2, \dots, y_i)$ with the property $x_i = r_i$ with probability $\frac{1}{2}$ and $x_i = s_i$ also with probability $\frac{1}{2}$. The same is true for the coefficients y_i.

Mutation of a quality control system is an operation during which the system $Q = (k_1, k_2, \dots, k_i)$ is transformed into $Q' = (k'_1, k'_2, \dots, k'_i)$ with the property that there is exactly one such index i for which $1 \le |k_i - k'_i| \le 3$. For the other indices $j \neq i$, however, there occurs $k_j = k'_j$. Thus, we can say that mutation involves changing the multiplicity of a single procedure/gene by one, two or three up or down.

The radius of a niche σ is that distance within which one looks for quality control systems that are in some sense similar to each other from the point of view of the NPGA algorithm. The distance of two systems itself is defined by a u-matching function. We will say that the systems Q_1 and Q_2 are distant by d if $d = |u(M_1) - u(M_2)|$. The radius of the niche should be chosen so that during the optimization process it can give a good indication of which control systems are so close that they are very similar to each other from the optimizer's point of view. The task of the NPGA algorithm is not to allow a situation in which the population has a lot of almost identical control systems – because then the complexity of the algorithm will increase a lot, but there will be no diversity of results.

Selecting appropriate parameters for mutation and crossover/recombination is crucial for the genetic algorithm's performance, as indicated by the extensive body of literature. Studies have shown that mutation helps maintain diversity within the population by exploring new areas of the solution space, while crossover combines the strengths of different individuals, leading to potentially better solutions. Specifically, Eiben et al. [20] and Deb [23] emphasize the role of mutation in avoiding premature convergence and the effectiveness of crossover in guiding the population toward optimal solutions. The careful tuning of these parameters can significantly affect the convergence rate and the quality of the solutions found by the algorithm.

NPGA algorithm diagram for quality control system optimizer

Algorithm input:

- number of quality control systems analyzed N,
- maximum number of generations T,
- quality control system fit function u (its weighting factors),
- cost function of the quality control system c,
- maximum budget for quality control system B,
- probability of crossover between quality control systems p_c,
- probability of mutation of quality control system p_m,
- count of the comparison set t_{dom}.

Algorithm output:

- a set of non-dominant quality control systems A.

Algorithm:

1. INITIATION
   Let $P_0 = \emptyset$ and $t = 0$. Then for $i = 1, \dots, N$ choose a random quality control system Q with such a property that $c(Q) < B$ and add it to the set P_0.
2. ADAPTATION AND SELECTION
   Set $P' = \emptyset$ and $i = 1$ and execute:
   a. For the Q_i system, randomize $Q_j \in P_t$ to compete and populate the P_{dom} comparison set with random individuals according to the value of t_{dom}.
   b. If the system Q_i matures over the set P_{dom} and the system M_j does not dominate over P_{dom} then Q_i is the winner of the tournament and is transferred to the provisional population $P' = P' + \{Q_i\}$.
   c. If, on the other hand, the system Q_j dominates the set P_{dom} and the system Q_i does not dominate P_{dom} then $P' = P' + \{Q_j\}$,
   d. If the conditions in the previous two points are not met then:

```
            i. calculate the number of systems in the temporary population whose distance from $Q_i$ is less
               than the radius of the niche, use the formula:
                           $n(Q_i) = |\{Q_k: Q_k \in P', |u(Q_i) - u(Q_k)| < \sigma\}|$
            ii. proceeding as above, calculate $n(Q_j)$,
            iii. if $n(Q_i) < n(Q_j)$ then $P' = P' + \{Q_i\}$ otherwise $P' = P' + \{Q_j\}$,
       e.   Set $i = i + 1$, if $i < N$ go to the first subsection, otherwise go to the next step.
3.  RECOMBINATION
            Set $P'' = \emptyset$ and for each $i = 1, \ldots, N/2$ perform:
       a.   Select two systems $Q_i, Q_j \in P'$ and then remove them from $P'$.
       b.   Cross $Q_i$ and $Q_j$ systems, resulting in descendant systems $Q_k$, $Q_l$.
       c.   Add $Q_k$, $Q_l$ to $P''$ with the probability $p_c$ otherwise move the systems $Q_i$, $Q_j$ do $P''$.
4.  MUTATION
            Set $P''' = \emptyset$ for each $Q_i \in P''$ perform:
       a.   Subject the $Q_i$ system to mutation with probability $p_m$, the mutation results in the $Q_j$ system.
       b.   Add system $Q_j$ to $P'''$.
5.  ENDING
```

Set a new base generation $P_{t+1} = P'''$ and $t = t + 1$. Check if $t \geq T$ or the additional stopping criterion is satisfied, place the non-dominated solutions from the population P_t in the set A and terminate the algorithm. Otherwise, proceed to the second step.

RESULTS

The algorithm presented in the previous section was tested using randomly generated test vectors. For this purpose, Python software was used along with the appropriate libraries for statistical and mathematical calculations [41, 42]. The program that generates the test vectors accepts a configuration file of the following form:

```
“configuration”: {
“depth”: 3,
“procedures”: {
“Q_1”: {
“T_1”: p_11,
...
“T_m”: p_1m
},
...
“Q_n”: {
“T_1”: p_n1,
...
“T_m”: p_nm
}
},
“series”: {
“T_1”: k_1,
...
“T_m”: k_m
}
}
```

In this file, the objects Q_i are labels of quality control procedures that can be used by the system. On the other hand, the objects T_j are labels of defect types that the system should recognize. It is clear that not every procedure can recognize any type of defect. Therefore, the configuration file specifies the probability p_ij with which the procedure Q_i is able to recognize a defect of type T_j. While the term probability is used here as a practical indicator of a system’s ability to detect defects, it should be noted that probability, in this context, does not represent a formal mathematical measure as defined in measure theory. Instead, it serves as an empirical estimate of performance, offering a quantitative way to model the likelihood of defect detection, which is critical in optimizing the system’s configuration. A summary of the properties of the procedures is included in the procedures object. The series object, in turn, is information on how many test vectors for a given defect type should be generated. The depth parameter, on the other hand, indicates how many times a procedure has been used to detect a defect.

A set of 5 procedures Q1, Q2, Q3, Q4, Q5 and 10 types of defects T1,..., T10 were prepared for the method convergence tests. The cost of each procedure was set at the same level so as not to distinguish any of them. During testing, the only optimization criterion was to maximize the probability of detecting a defect (this is how the fitting function u was determined). The budget of the quality control system was set so that it was impossible to use more than 3 procedures (multiple procedure use was allowed). The set of quality control procedures was divided into 4-element subsets. Then for each subset and each pair (*N, T*) from the set {3, 4,...,50} × {1, 2,...,50} 10 runs of the proposed version of the NPGA algorithm were performed. This made it possible to determine the averaged probability of defect detection for the basic parameters of the algorithm. Figure 1 shows the value of the *u* fit function. On the other hand, Figure 2 shows the discrepancy between the possible maximum value of the *u* function and the averaged values indicated by the NPGA algorithm. It should be noted that the limitations adopted for the purpose of conducting the experiments did not allow to obtain 100% efficiency in detecting the defect. Therefore, the actual efficiency of the algorithm, and therefore its convergence, should be read from the graphs in Figure 2. The smaller the value on them, the smaller the difference between the answer given by the NPGA and the sought optimal value.

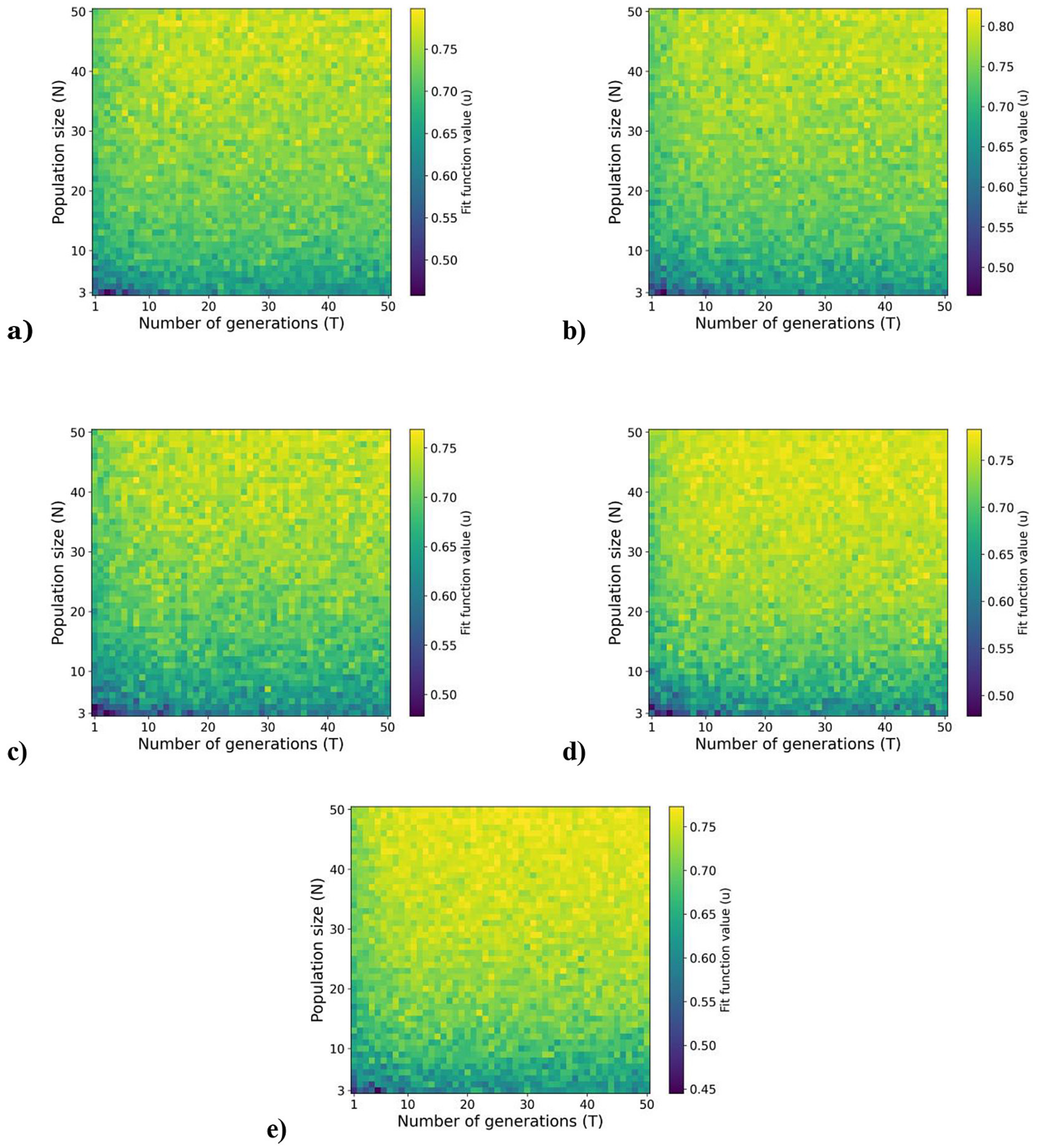

Figure 1. Graphs showing the values obtained by the presented version of the optimization algorithm as a function of the population size *N* and the number of generations *T* for subsets: a) {*Q1,Q2,Q3,Q4*}, b) {*Q1,Q2,Q3,Q5*}, c) {*Q1,Q2,Q4,Q5*}, d) {*Q1,Q3,Q4,Q5*}, e) {*Q2,Q3,Q4,Q5*}

To illustrate the impact of the recombination/crossover probability p_c and the mutation probability p_m on the optimization process, tests were conducted on a population with a fixed size of $N = 50$ individuals. The tests were performed on a subset of procedures Q2, Q3, Q4, Q5, which provided the best variation in results depending on the value of the $p_c$or p_m parameter. Figure 3a shows the discrepancy between the achievable maximum value of the function u and the averaged values indicated by the NPGA algorithm for different values of the p_c coefficient and varying numbers of generations T. The graph shows that the best convergence was obtained for values of p_c exceeding 0.5. This indicates that the recombination/crossover process is of significant importance to the final optimization result. Meanwhile, Figure 3b shows the discrepancy between the achievable maximum value of the function u and the averaged values indicated by the NPGA algorithm for different values of the p_m coefficient and varying numbers of generations T. In this case, it can be observed that the algorithm achieves the best results for values of $p_m \in (0.5; 0.7)$. Thus, the

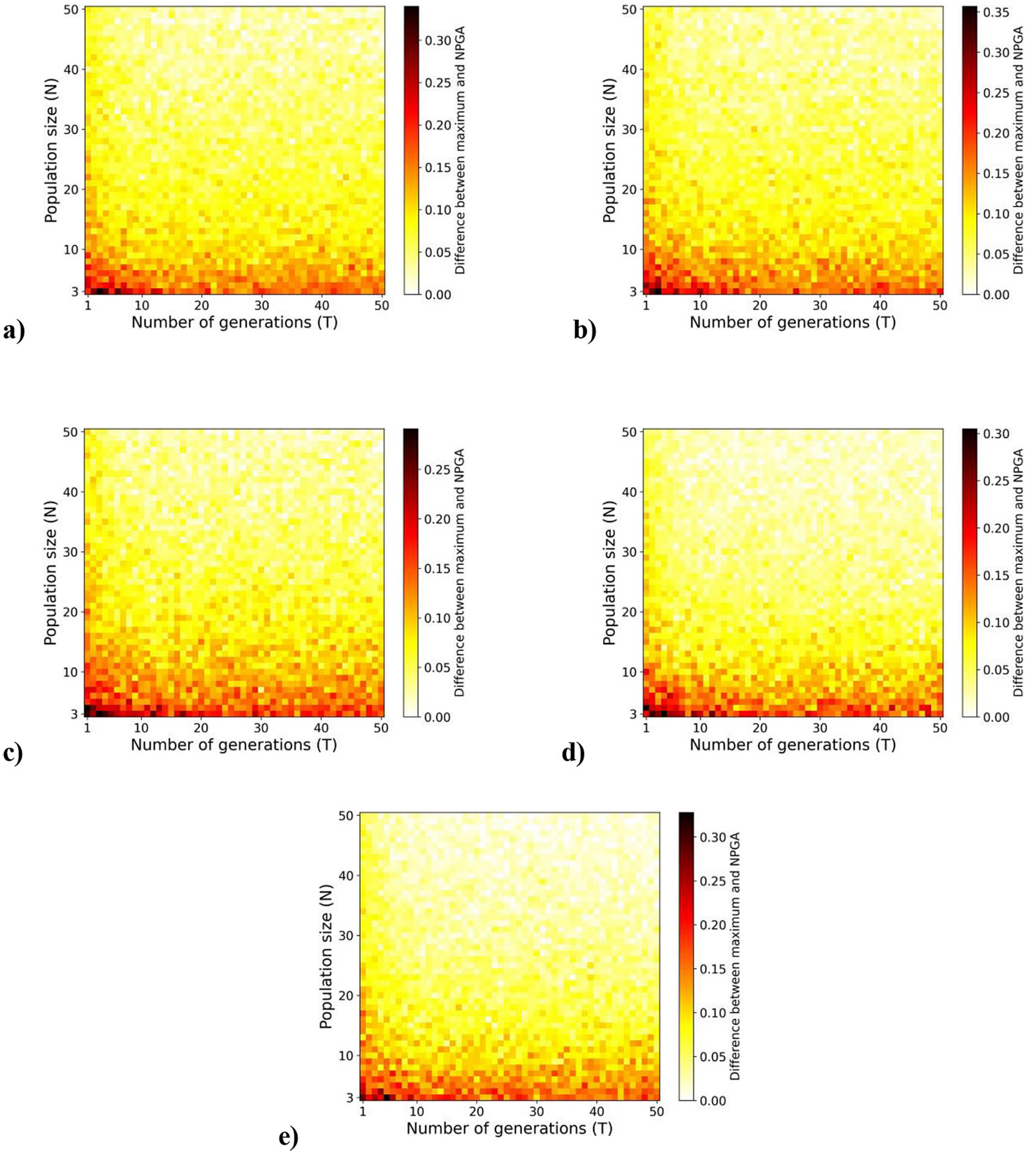

Figure 2. Graphs showing the discrepancy between the correct maximum and the values obtained with the presented optimization algorithm depending on the population size N and the number of generations T for the subsets: a) {Q1,Q2,Q3,Q4}, b) {*Q1,Q2,Q3,Q5*}, c) {*Q1,Q2,Q4,Q5*}, d) {*Q1,Q3,Q4,Q5*}, e) {*Q2,Q3,Q4,Q5*}

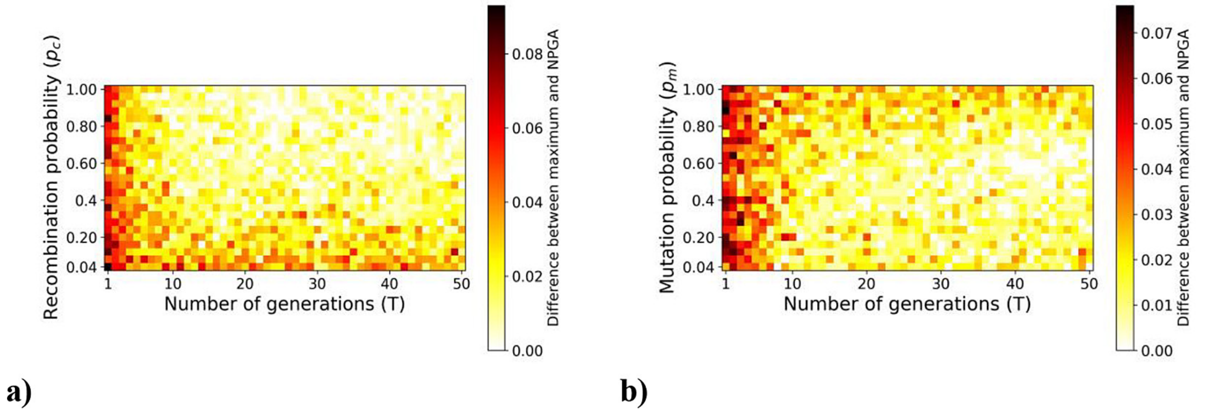

Figure 3. Graphs showing the discrepancy between the correct maximum and the values obtained using the presented optimization algorithm for a fixed population size $N = 50$, varying number of generations T, and varying parameter values: a) p_c representing the crossover/recombination probability, b) p_m representing the mutation probability, for the subset $\{Q2,Q3,Q4,Q5\}$

mutation process also plays an important role in the convergence of the optimization algorithm.

From the data presented, it is clear that a high efficiency of the algorithm can be achieved only with an adequate population size N. In contrast, the number of generations T plays a lesser role in the process of reaching the optimal value. The presented graphs show that very good results are possible even with 10 generations. On the other hand, even a very large number of generations does not allow to approach the optimal value if the population size is at a level lower than 30÷40 individuals.

These findings demonstrate that the NPGA is particularly effective when the population size is large enough to maintain diversity, which is critical for exploring the Pareto front efficiently. Compared to other optimization techniques, such as standard genetic algorithms or simpler multi-objective approaches, NPGA shows superior performance in avoiding premature convergence. This is primarily due to the niche pressure mechanism, which helps to maintain solution diversity and ensures that non-dominant solutions are more evenly distributed across the Pareto front.

The implications of these results are significant for industries where the cost and efficiency of quality control systems are critical. By applying NPGA, companies can achieve more robust optimization of their quality control processes, balancing the trade-off between cost and defect detection with greater precision than other methods allow. For example, compared to traditional Pareto-based algorithms, which may struggle with high-dimensional problems, NPGA provides a more scalable solution that can be adapted to complex, multi-objective environments.

In summary, this research highlights the novelty of NPGA in optimizing quality control systems. It fills a gap in the literature by demonstrating how this algorithm can outperform existing techniques in multi-objective optimization tasks, particularly when it comes to maintaining diversity in the solution space and avoiding suboptimal convergence. Further comparisons with other approaches, such as hybrid algorithms combining genetic methods with neural networks, could provide additional insights into the specific advantages of NPGA in various application domains.

CONCLUSIONS

Quality control systems are essential for ensuring high-quality products and services, yet they often incur significant costs related to hardware, software, human resources, and other operational expenses. Optimizing these costs while maintaining system efficiency is a significant challenge for companies. Genetic algorithms, inspired by natural evolutionary processes, provide a promising approach to addressing this challenge. They utilize mechanisms such as mutation, crossover, and selection to find optimal or near-optimal solutions to complex optimization problems.

Convergence tests of the proposed NPGA method highlighted that its effectiveness in maximizing the probability of defect detection is significantly influenced by the population size. The experiments demonstrated that high performance can be achieved with a relatively small number of generations, around 10, provided the population size is sufficiently large, specifically at least 30 to 40 individuals. Conversely, a high number of generations cannot compensate for a small population size, underscoring the importance of maintaining an adequate population size for optimal results.

However, the application of genetic algorithms comes with challenges. Properly defining the objective function and constraints is crucial for satisfactory results. Additionally, selecting appropriate parameters for the genetic algorithm, such as population size and mutation and crossover probabilities, is essential for efficient and effective optimization.

Despite these promising results, the study has several limitations that should be addressed in future research. One limitation is the reliance on a predefined set of test vectors and procedures, which may not fully capture the variability of real-world quality control environments. Future studies could extend this work by testing the NPGA algorithm in more diverse industrial scenarios or applying it to dynamically changing environments where quality control processes evolve over time. Another area for further investigation is the exploration of hybrid approaches, combining NPGA with other optimization techniques or machine learning methods to improve its adaptability and robustness.

Moreover, while this study provides a clear demonstration of NPGA's effectiveness, it is essential to highlight the novel contributions of this research. This work advances the existing body of knowledge by being one of the first to apply the NPGA in the context of quality control optimization. Its use of niche pressure to maintain solution diversity represents a significant improvement over traditional Pareto-based algorithms, which often struggle with high-dimensional multi-objective problems. This innovation ensures a more comprehensive search of the solution space, resulting in better overall optimization performance.

Also noteworthy is the innovative approach to representing genes as probability-generating functions. This approach opens up new possibilities for utilizing the NPGA algorithm in situations where a process or service can be represented as a stochastic variable approximated by appropriate statistics.

In conclusion, genetic algorithms offer a powerful tool for optimizing quality control systems by balancing cost and efficiency. Their effectiveness, particularly in complex, multi-criteria optimization problems, can be significantly enhanced through careful parameter selection and innovative approaches like the Niched Pareto MOGA. This ensures a comprehensive and representative set of optimal solutions, thereby improving the overall performance and cost-efficiency of quality control systems. By addressing the limitations mentioned and exploring further applications of NPGA, future research can unlock even greater potential for this approach in both quality control and other industrial optimization problems

REFERENCES

1. Baranovskyi D., Myamlin S., Bulakh M., Podosonov D., Muradian L. Determination of the filler concentration of the composite tape. Applied Sciences. 2022, 12(21), 11044. https://doi.org/10.3390/app122111044
2. Baranovskyi D. and Myamlin S. The criterion of development of processes of the selforganization of subsystems of the second level in tribosystems of diesel engine. Scientific Reports. 2023, 13(1), 5736. https://doi.org/10.1038/s41598-023-33015-3
3. Baranovskyi D., Bulakh M., Michajłyszyn A., Myamlin S., Muradian L. Determination of the risk of failures of locomotive diesel engines in maintenance. Energies. 2023, 16, 4995. https://doi.org/10.3390/en16134995
4. Daneshjo N, Malega P., Drábik P. Techniques for production quality control in the global company. Advances in Science and Technology Research Journal. 2021, 15(1), 174–83. https://doi.org/10.12913/22998624/131558
5. Oborski P. and Wysocki P. Intelligent visual quality control system based on convolutional neural networks for holonic shop floor control of industry 4.0 manufacturing systems. Advances in Science and Technology Research Journal. 2022, 16(2), 89–98. https://doi.org/10.12913/22998624/145503
6. Nikolay K., Mikhail O., Nikolay B., Oleg F., Hayakawa M. Various approaches to problems of multicriterion optimization processes of electric power systems. In Proceedings of 2014 International Symoposium On Electromagnetic Compatibility, Tokyo (EMC'14/Tokyo), 2014, 418–421.
7. Al-Zuheri A. Cross-comparison of evolutionary algorithms for optimizing design of sustainable supply chain network under disruption risks. Advances in Science and Technology Research Journal. 2021, 15(4), 342–351. https://doi.org/10.12913/22998624/142213
8. Iwańkowicz R.R. and Sekulski Z. Solving the

problem of vehicle routing by evolutionary algorithm. Advances in Science and Technology Research Journal. 2016, 10(29), 97–108. https://doi.org/10.12913/22998624/61937

9. Kaya I. A genetic algorithm approach to determine the sample size for attribute control charts. Information Sciences. 2009, 179(10), 1552–1566. https://doi.org/10.1016/j.ins.2008.09.024
10. Hatjimihail A.T. Genetic algorithms-based design and optimization of statistical quality-control procedures. Clinical Chemistry. 1993, 39(9), 1972–1978. https://doi.org/10.1093/clinchem/39.9.1972
11. Amin M., Ghaly A., Ayad F., Hosny O. Overall schedule optimization using genetic algorithms. In Proceedings of the Canadian Society of Civil Engineering Annual Conference. 2024, 449–461. https://doi.org/10.1007/978-3-031-35471-733
12. Hinojosa M.A., López-Sánchez A.D., Hernández-Díaz A.G., Santana-Quintero L.V. Compromise utilitarian solutions in multi-criteria optimization problems as a guide for evolutionary algorithms. Computers and Operations Research. 2013, 40(5), 1155–1164.
13. Goldberg D.E. Genetic algorithms in search, optimization and machine learning. Addison Wesley Longman Publishing Co., Inc., 1989, USA.
14. Chambers L. Practical Handbook of Genetic Algorithms: New Frontiers. CRC-Press. 2, 1995. https://doi.org/10.1201/9780429128332
15. Zhou Q., Wan Y., Wong W.K., Loh P.L., Lee Y.C. Semiconductor integrated circuit product quality improvement utilizing genetic algorithm-based optimization technique. Journal of Intelligent Manufacturing, 2009, 20(1), 83–115.
16. Duffuaa S.O., Khan M., Daya B. An exploration of the impact of implementing jit and ie techniques for manufacturing quality systems. International Journal of Six Sigma and Competitive Advantage, 2004, 1(1), 95–118.
17. Chen Y.S. and Tsai C.H. An optimization approach to multi-characteristic quality control using genetic algorithms. International Journal of Advanced Manufacturing Technology. 2009, 40(7–8), 693–709.
18. Sadough Vanini Z.N., Mohammadi K., Khorshidi H.A. A hybrid approach of fuzzy genetic algorithm for optimization of maintenance plan in a quality control system. International Journal of Quality & Reliability Management, 2015, 32(3), 248–268.
19. Coello Coello C.A. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering. 2002, 191(11–12), 1245–1287. https://doi.org/10.1016/S0045-7825(01)00323-1
20. Eiben A.E., Hinterding R., Michalewicz Z. Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation. 1999, 3(2), 124–141. https://doi.org/10.1109/4235.771166
21. Von Lücken C., Baran B., Brizuela C.A. A survey on multi-objective evolutionary algorithms for many-objective problems. Computational Optimization and Applications. 2014, 58(3), 707–756. https://doi.org/10.1007/s10589-014-9644-1
22. Coello Coello C.A., Lamont G.B., Van Veldhuizen D.A. Evolutionary algorithms for solving multi-objective problems. Springer, 2007, New York. https://doi.org/10.1007/978-0-387-36797-2
23. Deb K. Multiobjective optimization using evolutionary algorithms. Wiley, New York, 2001.
24. Corne D.W. and Knowles J.D. Techniques for highly multiobjective optimisation: Some nondominated points are better than others. In: Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference, New York, Association for Computing Machinery, 773–780, https://doi.org/10.48550/arXiv.0908.3025
25. Farina M. and Amato P. On the optimal solution definition for many-criteria optimization problems. In: Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society. NAFIPS-FLINT. New Orleans, LA, USA, IEEE 2002, 233–238. https://doi.org/10.1109/NAFIPS.2002.1018061
26. Hughes E. Evolutionary many-objective optimisation: Many once or one many? In: 2005 IEEE Congress on Evolutionary Computation, Proceedings, 2005, 1, 222–227. https://doi.org/10.1109/CEC.2005.1554688
27. Khare V., Yao X., Deb K. Performance scaling of multi-objective evolutionary algorithms. In: Evolutionary Multi-Criterion Optimization, Berlin, Heidelberg, Springer, 2003, 376–390.
28. Knowles J.D and Corne D.W. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Lecture Notes in Computer Science. Berlin, Heidelberg, Springer, 757–771. https://doi.org/10.1007/978-3-540-70928-2 57
29. Purshouse R.C. and Fleming P.J. Evolutionary many-objective optimisation: an exploratory analysis. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003). 2066–2073. https://doi.org/10.1109/CEC.2003.1299927
30. Purshouse R.C. and Fleming P.J. On the evolutionary optimization of many conflicting objectives. IEEE Transactions on Evolutionary Computation, 2007, 11, 770–784. https://doi.org/10.1109/CEC.2003.1299927
31. Ishibuchi H., Tsukamoto N., Nojima Y. Evolutionary

many-objective optimization: A short review. IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, 2419–2426. https://doi.org/10.1109/CEC.2008.4631121

32. López Jaimes. A., Coello Coello C.A., Urías Barrientos A.E. Online objective reduction to deal with many-objective problems. In: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science, Berlin, Heidelberg. Springer. 423–437. https://doi.org/10.1007/978-3-642-01020-0 34
33. Fonseca C.M. and Fleming P.J. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation, 1995, 3, 1–16. https://doi.org/10.1162/evco.1995.3.1.1
34. Mostaghim S., Branke J., Schmeck H. Multi-objective particle swarm optimization on computer grids. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO ’07), New York, ACM Press, 2007, 869–875. https://doi.org/10.1145/1276958.1277127
35. Obayashi S. and Sasaki D. Visualization and data mining of pareto solutions using self-organizing map. In: Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Lecture Notes in Computer Science, Berlin, Springer, 796–809. https://doi.org/10.1007/3-540-36970-8 56
36. Pryke A., Mostaghim S., Nazemi A. Heatmap visualization of population based multi objective algorithms. In: Proceedings of the 4th International Conference on Evolutionary MultiCriterion Optimization (EMO 2007). Lecture Notes in Computer Science, Berlin, Heidelberg, Springer, 361–375. https://doi.org/10.1007/978-3-540-70928-2 29
37. Walker D., Fieldsend J., Everson R. Visualising many-objective populations. In: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference Companion (GECCO Companion ’12), New York, USA. Association for Computing Machinery, 2012, 451–458. https://doi.org/10.1145/2330784.2330853
38. Baraldi P., Pedroni N., Zio E. Application of a niched pareto genetic algorithm for selecting features for nuclear transients’ classification. International Journal of Intelligent Systems, 2009, 24, 118–151. https://doi.org/10.1002/int.20328
39. Zitzler E. Evolutionary algorithms for multiobjective optimization: methods and applications. Swiss Federal Institute of Technology, Zurich, 1999.
40. Horn J.D., Nafpliotis N., Goldberg D.E. A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 1994, 82–87.
41. Chmielowiec A. and Klich L. Application of python libraries for variance, normal distribution and weibull distribution analysis in diagnosing and operating production systems. Diagnostyka, 2021, 22(4), 89–105. https://doi.org/10.29354/diag/144479.
42. Chmielowiec A. Algorithm for error-free determination of the variance of all contiguous subsequences and fixed-length contiguous subsequences for a sequence of industrial measurement data. Computational Statistics, 2021, 36(4), 2813–2840. https://doi.org/10.1007/s00180-021-01096-1