Applied Soft Computing Journal 161 (2024) 111372

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

On the use of the differential evolution algorithm for truss-type structures optimization

Oscar Contreras-Bejarano [a,*], Jesús Daniel Villalba-Morales [b]

[a] Department of Structural and Geotechnical Engineering, Pontificia Universidad Católica de Chile, Santiago de Chile, Chile
[b] Department of Civil Engineering, Pontificia Universidad Javeriana, Bogotá, Colombia

ARTICLE INFO

Keywords:
Differential evolution algorithm
Problem knowledge heuristics
Multi-modal optimization
Parameter control
Truss optimization

ABSTRACT

In recent decades, bio-inspired numerical algorithms have emerged as an alternative for optimizing the structural design of trusses. The differential evolution algorithm (DEA) has demonstrated both good performance and ease of implementation. However, unlocking the full potential of DEA to address engineering problems poses a significant challenge, necessitating a strategic and informed definition of each component of the algorithm. This research systematically evaluates the influence of defining DEA components on improving the reliability of truss optimization. The algorithm structure of DEA was configured for five aspects: (I) the mutation operator, (II) inclusion of multi-modal techniques, (III) inclusion of parameter control techniques, (IV) definition of the initial population, and (V) local search heuristics. A comprehensive evaluation is conducted to assess the performance of 23 DEA configurations in optimizing eight planar and spatial trusses, varying in size from 10 to 163 elements. Assessment is based on key criteria such as optimal weight, robustness, and computational cost, providing a thorough basis for comparison. The results showed that no tested DEA configuration is the best for all trusses. Instead, the study revealed the presence of recommendable configurations, each tailored to the specific complexities and scales inherent in various truss structures. The integration of multimodal and local search techniques proves particularly advantageous for larger trusses, amplifying the algorithm's exploratory capabilities to effectively navigate and uncover optimal regions. In contrast, using parameter control technique was more effective in optimizing smaller trusses, capitalizing on the rapid exploration of potential optimal areas in a smaller search space. There was a mutation operator that produced the best results for large trusses and good results for smaller structures. This operator uses the target vector as the base vector and guides its movement from a best-based difference vector, achieving a balance between the exploratory stage and user-defined constraints that promote the exploitation of potentially optimal areas. No discernible impact was observed when the initial population heuristic proposed was used. Finally, this study underscores the feasibility of DEA configurations for optimizing trusses of varying complexities, as proved by a comparison with results from the literature.

1. Introduction

Truss-type structural systems are widely used to build civil infrastructure such as roofs and bridges. The cost of these structures is directly tied to their importance and size, with the final price depending on the cost of the materials and the construction process. Consequently, the benefits of employing optimization techniques to minimize the cost of structural design become more pronounced as the importance or size of the truss increases. Notable examples of optimized structural examples include a planar steel truss arch bridge [1], the CIGRÉ line transmission tower [2], a cold-formed steel residential roof truss [3], a stacker crane [4] and a structural dome [5]. The significance of cost savings becomes particularly apparent when there is a need to produce numerous structures. For example, the implementation of optimization techniques can lead to a reduction of approximately 21% in the manufacturing cost of certain electrical towers [6]. Furthermore, the integration of optimization into the construction industry has a positive environmental impact by reducing the amount of materials used.

At the lowest level of numerical complexity, truss optimization (TrO) involves determining the cross-section of the bars to achieve a structure with the minimum cost. The associated mathematical problem can be formulated in several ways, depending on the type of design variable, the objective function, and the constraints. Design variables can be addressed mainly through discrete (utilizing a cross-section database available) and continuous (utilizing a range of area values)

* Corresponding author.
E-mail addresses: ocontreras1@uc.cl (O. Contreras-Bejarano), jesus.villalba@javeriana.edu.co (J.D. Villalba-Morales).

https://doi.org/10.1016/j.asoc.2024.111372
Received 21 August 2023; Received in revised form 23 January 2024; Accepted 4 February 2024
Available online 8 February 2024

representations. In many research efforts, the same cross-section is assigned to groups of truss elements to consider constructive issues, thereby reducing the complexity of the search space due to a lower number of design variables [7]. Regarding the objective function, it is commonly accepted that the total weight of the structures serves as a quality measure for the structural design [8–11]. Typically, constraints consider functional issues related to the displacement and stress of nodes in the elements. But other limitations could be based on factors such as natural frequency [12], construction planning [13], buckling [14], global stability [15] and imperfections [14]. Furthermore, there exists the possibility of using a multi-objective approach, as demonstrated in [16], where the maximum displacement and the mass of the truss are used. In summary, the formulation of the optimization problem requires a precise definition of these mentioned aspects.

Over the last decade, meta-heuristics have gained importance in solving the TrO problem due to the results reported in the literature and their straightforward computational implementation [17–19]. However, the success of these approaches depends on the correct selection and configuration of the algorithm, which implies a better understanding between the search operators and the characteristics of the objective space. Regardless of the metaheuristic, the convergence process can be improved by knowledge-based heuristic rules, mainly considering structural responses such as displacement, buckling [20], and stress [21]. Furthermore, the performance of a metaheuristic can be augmented through various strategies, such as the inclusion of optimality criteria to guide the search [18], special definitions of the initial population to locate the promising regions in the search space [22], the reduction of the complexity level [23], and the hybridization with other algorithms [24]. Algorithm parameters affect exploration and exploitation in the search space, but it is possible to employ parameter control techniques to define them. The idea is to allow for a dynamic definition during the search without user interaction, as did Xie et al. [25]. The multimodal nature of the TrO problem adds complexity to the achievement of the optimal solution [26], indicating the need to equip the algorithm with computational routines to deal with it. In structures with numerous variables, the computational cost becomes a significant consideration. To mitigate this, strategies such as metamodels [27], reanalysis methods [28], parallel metaheuristic Algorithms [29], or multilevel optimization [30] can be employed. Furthermore, metaheuristics can be applied to solve the multi-objective approach for the TrO problem, as in [31–33]. Taking into account the above issues in the solution of the TrO problem, it is possible to ensure the algorithm's effectiveness across a broad spectrum of trusses studied.

Exploring the feasibility of implementing the Differential Evolution Algorithm (DEA) for the TrO problem is crucial, given its established track record in tackling diverse optimization challenges. DEA employs a stochastic procedure, iteratively modifying a set of solutions through evolutionary operators such as mutation, crossover, and selection [34]. It has shown good performance in solving various types of structural optimization, including size [35], material [36], shape and topology optimization [37]. Its versatility extends to various domains of structural engineering, including structural damage detection [38], optimal location of sensors for structural health monitoring [39], continuous steel casting production [40], bridge maintenance plans [41], vertical design spectra [42], structural reliability analysis [43], identification of optimal parameters in concrete [44] and optimal location of viscous dampers in structures [45]. It is worth mentioning that some theoretical developments support the availability of DEA for optimization [46].

Numerous studies highlight the potential effectiveness of the DEA in optimizing the structural design of trusses. Yi et al. [47] utilize a dynamic self-adaptive approach, adjusting DEA parameters to generate offspring and achieving a more balanced blend of algorithmic exploitation and exploration capabilities. Lemonge et al. [48] introduces multi-objective structural optimization problems by combining novel conflicting objective functions and constraints. These include considerations for natural frequencies of vibration and load factors related to the global stability of the structure. Kao et al. [34] present two modifications to the mutation strategy. The first integrates three mutation operators to enhance diversity, while the second adapts the mutation factor across five distinct optimization problems concerning the weight of the truss structures. Similarly, Deng et al. [49] introduced a DEA approach that focuses on improving the relationship between the convergence rate and the balance of global and local search aspects. Zeng et al. [50] Developed a selection strategy utilizing discarded trial vectors to enhance the performance of the DEA. Chakraborty et al. [51] compare the performance of multiple DEA variants. Upon performance analysis, it is revealed that none of the Differential Evolution (DE) variants can efficiently solve all the given problems. Charalampakis and Tsiatas [52] corroborate these findings, reporting that DEA exhibits superior performance compared to other metaheuristics. Several other references on the application of DEA to TrO can recently be found in the literature [29,37,53–61]. The number of DEA approaches reported in the literature allows us to think that the best DEA configuration for the TrO problem has not been configured yet.

This article defines a configuration scheme for the original DEA [62], assessing the performance of various algorithm operators, that allows one to get desirable results to solve the truss optimization problem. Five issues are analyzed to understand the DEA's success in exploring and exploiting the search space by (1) comparing twelve mutation functions, (2) overcoming the multimodality of the objective function considering three multimodal techniques, (3) evaluating the impact of a parameter control strategy in the process, (4) defining an initial population heuristic rule, and (5) testing six local search procedures. The study encompasses the analysis of eight trusses, exploring a total of 23 algorithm configurations, resulting in 184 detailed analyzes. For each characteristic analyzed, the heuristic rule yielding the optimal truss is identified and employed to establish a performance ranking of the heuristics. Subsequently, the performance of the best DEA configuration is compared with the results from the literature. This research is significant for the following reasons:

- The study addresses a crucial need by systematically evaluating how the definition of various DEA components affects its overall performance in truss optimization. Examples of decisions that the user must make include the selection of the mutation operator and the incorporation of a multimodal technique.
- This research provides tailored insights into optimizing the structural design of trusses of varying complexity, offering nuanced recommendations specific to each truss type. This contributes to a more refined approach in engineering applications.
- The tested DEA configurations provide adequate results for trusses with a varying range of elements, as shown in a comparison with results from the literature.
- This paper describes the DEA concepts linked to their application in the truss optimization problem with the aim of facilitating the understanding of the process and the results.

2. Formulation of truss optimization problem

The mathematical complexity of the TrO problem varies depending on the available formulations. This research follows the standard formulation, which involves determining the optimal cross-sections for the elements of a specific truss. It is worth highlighting that this approach is similar to planar or spatial trusses. The study explores two cases based on the number of design variables: (1) one variable per group of bars assigned with the same section and (2) one variable per bar on the truss. The first case stems from the constructive necessity of having a few different cross-sections for the truss. Likewise, two cases arose concerning the cross-section definition: (1) continuous representation from a minimum to a maximum area and (2) discrete representation from a predefined allowed cross-section database. Option 2 aligns better with real-world situations. The geometric characteristics of the truss, applied

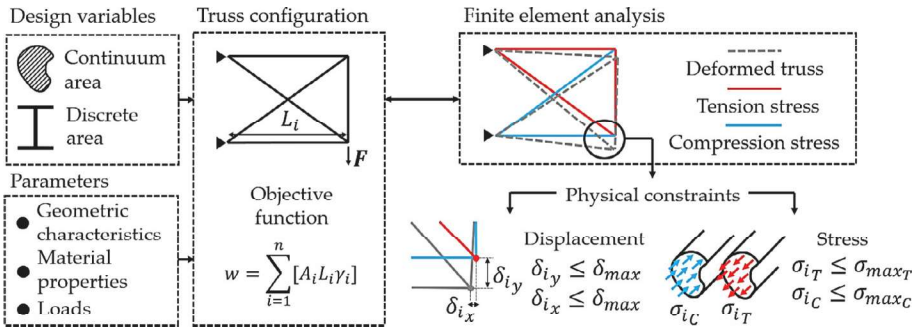

Fig. 1. Graphical formulation of the TrO problem.

loads, and material properties are treated as design parameters, with their values remaining constant throughout the optimization process. As an approximation, the quality of a structural design can be assessed from the calculation of the weight of the truss. At the same time, constraint validation involves computing maximum stresses in the elements and maximum nodal displacements. Fig. 1 shows a truss configuration to visually understand the TrO problem.

In the following, the truss optimization problem is formulated mathematically. The objective function corresponds to the minimization of the weight of the truss, as given by:

$$Minimize\ w = \sum_{i=1}^{n} \left[A_i L_i \gamma_i\right] \tag{1}$$

where w is the weight of the truss, A_i is the cross-sectional area of bar i (design variables), L_i the length of bar i, n is the number of bars and γ_i is the specific weight of each bar i. The problem is subjected to:

$$g_1 : \left[A_1, L_i\right] > 0 \tag{2}$$

$$g_2 : \delta_{max} - \delta_i > 0 \tag{3}$$

$$g_3 : \sigma_{max} - \sigma_i > 0 \tag{4}$$

where δ_{max} is the nodal maximum displacement of node i, and σ_{max} is the maximum stress in tension or compression in bar i. Typically, solving the TrO problem using metaheuristics involves transforming it into an unconstrained one through a penalization function. In this research, a static function was used as expressed by

$$W_R = W\left(1 + \alpha \frac{NNSC}{TNC}\right)^{\beta} \tag{5}$$

where W_R corresponds to the penalized weight based on the unfulfilled constraints, $NNSC$ is the Number of Not Satisfied Constraints and TNC is the Total Number of Constraints. Finally, α and β are initial parameters that the user must calibrate depending on the optimization problem.

3. Differential evolution algorithm

3.1. The physical analogy for DEA mathematical basis

DEA belongs to the category of optimization techniques known as evolutionary algorithms, drawing inspiration from Darwinian theory to establish their mathematical foundations. This technique is recognized for its ease of implementation on computers and its ability to deal with complex non-linear optimization problems [63]. The original DEA was proposed by Prince and Storn [62] according to the principle of evolution of a set (population) of solutions (vectors) to find the optimal solution to a given problem. The underlying idea is to iteratively enhance the population by applying operators that simulate mutation, crossover, and selection of dominant individuals. DEA finds applications in various areas of knowledge, with the TrO problem being one example.

The computational process of the DEA is described in Algorithm 1, beginning with the establishment of the initial population. The size of the population (NP) is defined according to the size of the problem, usually following the guideline provided in [62] of five to ten times the number of variables. Subsequently, a new population, referred to as the trial population, is generated from the existing one using two operators. The mutation operator selects n base vector and modifies them by incorporating characteristics from other vectors within the population. Mutant vectors may undergo a slight adjustment through a crossover operator to enhance recombination. These operators collectively contribute to exploring new possibilities, thus promoting the enhancement of the overall population. Then, a tournament is held between the trial and the current population, utilizing the objective function (fitness) as a metric for quality. Solutions exhibiting the highest fitness become the new parents in the next generation (iteration) of a maximization problem. Consequently, new individuals with lower skills face low survival prospects in the selection process for the next generation, ensuring the retention of the most proficient individuals. The process of obtaining new populations is repeated until the algorithm reaches a convergence to a solution. The application of DEA requires the definition of three parameters: (1) population size, (2) crossing rate (CR) and (3) scale factor (F). These parameters condition the reproduction process of individuals [64] that gradually move to an optimal potential space [65].

Step 1. Generate a uniformly distributed random initial population $(X^0_{i,j})$;

for $i = 1 : NP$ **do**

Step 2. Generate the mutated population $(V_{i,j})$

$$V_{i,j} = X_{Best} + F(X_{r1} - X_{r2}) \tag{6}$$

Step 3. Generate the crossover population $(U_{i,j})$

$$U_{i,j} = \begin{cases} V_{i,j}, & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand}. \\ X_{i,j}, & \text{otherwise.} \end{cases} \tag{7}$$

Step 4. Select the best population for the next generation

$$X^{g+1}_{i,j} = \begin{cases} U_{i,j}, & \text{if } f(U_{i,j}) < f(X_{i,j}). \\ X_{i,j}, & \text{otherwise.} \end{cases} \tag{8}$$

end

Step 5. If the condition is achieved, finish the algorithm, otherwise comeback to **step 2**

Algorithm 1: Pseudo code of Differential Evolution Algorithm

Fig. 2 illustrates the process of DEA utilizing the natural evolution of fish vision, transitioning from a blurred to a clearer underwater vision in successive generations. In the depicted fish shoal, a red-colored fish presents clear vision, while a purple-colored fish presents blurred vision. The clarity of vision corresponds to the quality measure, observing

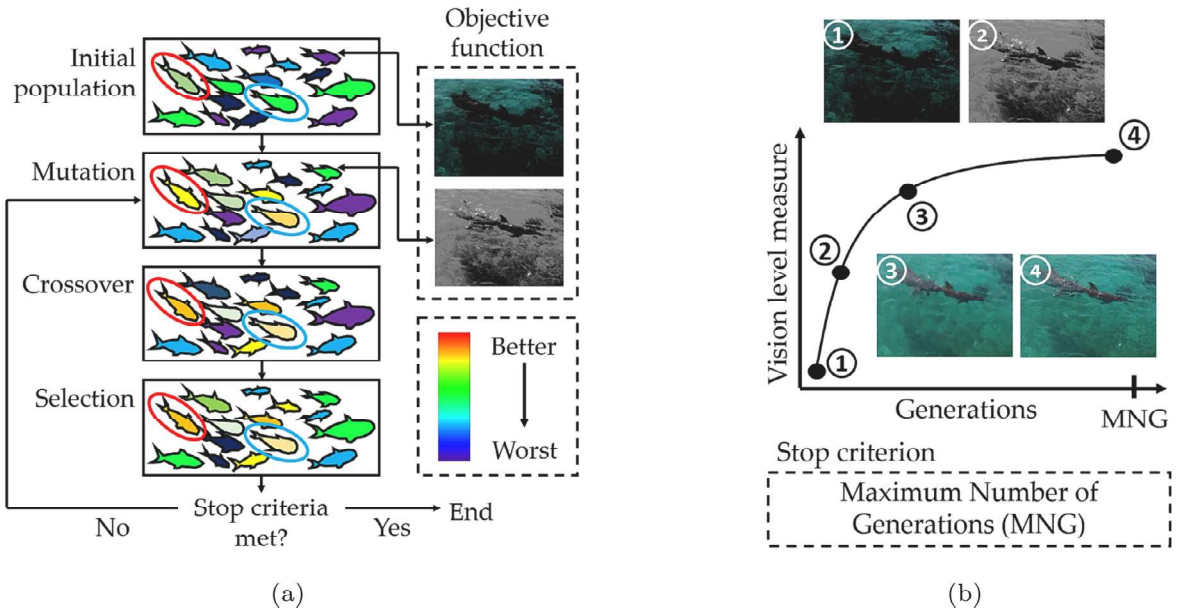

Fig. 2. Analogy of the process of evolution of the vision fish with DEA (a) principals steps and (b) convergence.

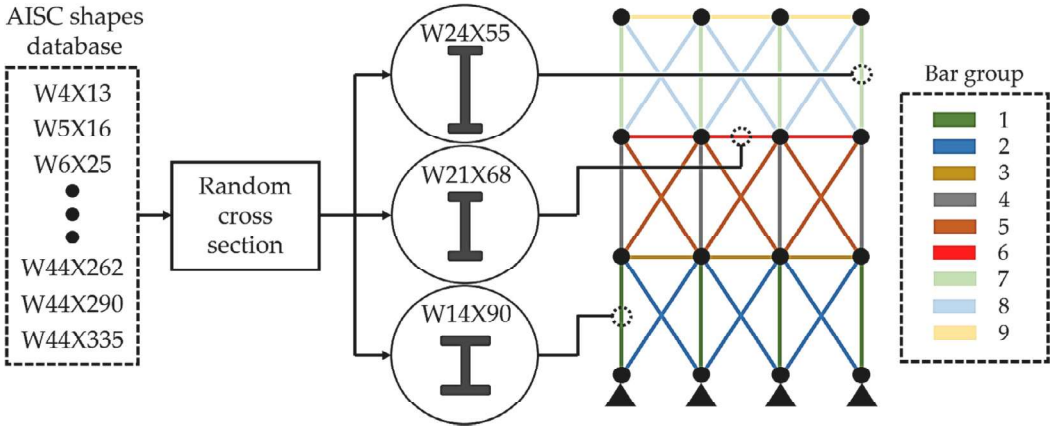

Fig. 3. Example of bar groups definition for a 2D truss.

that the initial population does not have a clear vision. When the mutation and the crossover operator are applied, the new shoal presents fishes with better vision, standing out the presence of an orange fish with good vision. Then, the current and trial populations coexist, but only the fish with a clearer view survive for the following generation. For example, purple fish faced extinction during the selection stage. At this point, the first evolution has been carried out. As the entire shoal does not yet exhibit clear vision, it becomes imperative to continue with the evolution process. Fig. 2 b shows the improvement in shoal vision across generations, revealing a significant improvement in the initial generations, with most fish having clearer vision at the conclusion of the process.

3.2. DEA and the TrO problem

In the following paragraphs, the stages of the DEA are explained in the context of the TrO problem. Population-based meta-heuristics requires defining the initial set of probable solutions for the problem, commonly achieved by generating random values for the design variables within a specified range. For the TrO problem, this range corresponds to the minimum and maximum areas in the cross-section database. Fig. 3 illustrates a possible configuration for the truss, considering the cross-sections available in the AISC shapes database. The elements in the truss are organized into nine bar groups and each is assigned a specific cross-section. The defined configuration falls into one of three cases: (i) an optimal solution, (ii) an unfeasible solution that does not satisfy at least one constraint, or (iii) an overdesigned truss.

Typically, structural designers opt to group elements with the same cross-section to streamline the construction process and minimize costs associated with fabrication. The criteria for group selection often involve subjective considerations, but are related to the constructive process and bars with similar stress. To demonstrate the impact of the grouping strategy on the optimization process, we assume the application of both random and standard strategies to a 72-bar truss (refer to Fig. 4). The standard grouping was carried out according to the existing literature, while the four random groups were generated by assigning each bar to a random group. The analysis reveals that the definition of elements in each group impacts the optimal solution, and a truss with a lower weight can be achieved if each element has its own section. Introducing randomness in group selection for bars led to varying cross-sections among similar bars, consequently yielding different weights for the truss. It is worth mentioning that it would be possible to configure an optimization process to define element groups, reducing the possible bias in the user definition.

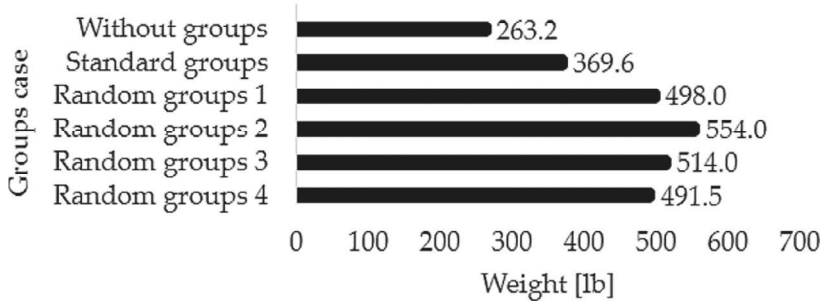

Fig. 4. Optimal weights for standard and random configuration of groups.

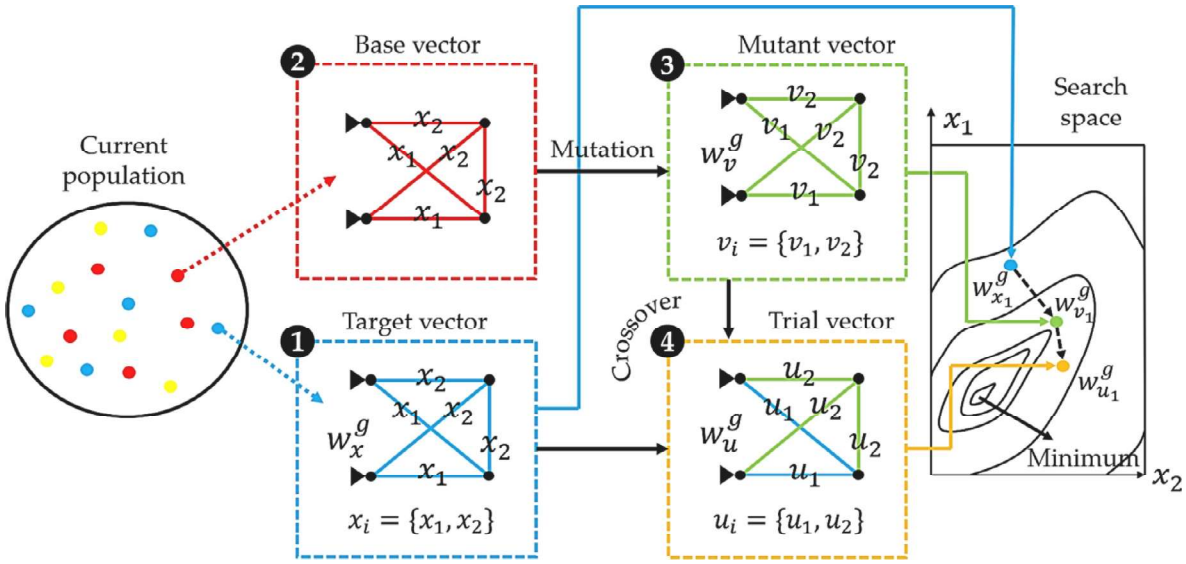

Fig. 5. Graphic process of mutation and crossover operator.

Another strategy to generate the population involves the use of problem-knowledge-based heuristics. The definition of population size depends on the dimension of the problem, specifically tied to the number of groups of elements in the TrO problem. Typically, this value remains constant throughout the optimization process and is determined through trial and error, involving the optimization of the truss for various population sizes. In the case of the TrO problem, the population size should be as low as possible due to the computational cost involved in finite element analysis for a truss configuration.

Fig. 5 resumes the application of mutation and crossover operators for a single target solution (Step 1). In the second step, a random base vector is chosen and modified by incorporating information from other trusses in the population, resulting in a new truss known as the mutant vector (Step 3). The green truss is then combined with the blue truss to generate a trial solution during the crossover phase (Step 4). One of the most commonly used crossover operators involves generating a 0–1 random number for each design variable. The trial vector assumes the value of the mutant vector if the random number is less than the crossover rate (CR); otherwise, it adopts the value of the target vector. In this study, CR equals 0.8 [66]. The primary objective of mutation and crossover operators is to generate new solutions derived from current ones, facilitating exploration of the search space. The right side of Fig. 5 indicates the positions of the different truss configurations in the search space, indicating that the trial vector could have moved to a better position.

Finally, a competition between the trial and the target trusses is carried out in the selection phase. The weights of both trusses are compared, and the one with the lowest value is chosen to be part of the next generation. This process is replicated for all the trusses in the population, one by one. Alternatively, another criterion involves considering the similarity between individuals; that is, a truss of the current population can only be compared with a similar truss of the trial population. The reason for this election is to preserve diversity in future generations. The convergence graph in Fig. 6 shows that the best initial truss configuration has a weight that is far from the optimal solution. As subsequent generations unfold, the optimal truss configuration gradually converges, demonstrating a minor improvement rate towards the conclusion of the evolutionary process.

3.3. Describing mutation operators

The importance of the mutation operator definition is focused on its impact on the algorithm's local and global search capabilities. With numerous mutation operators documented in the literature, the following question arises: What would be the best mutation operator to use in the case of TrO? This paper addresses this question by analyzing twelve mutation operators that have demonstrated strong performance in solving continuous optimization problems. Given the current lack of complete understanding regarding the relationship between the TrO objective function and the search characteristics of DEA, the next step involves testing the best mutation operators identified in the literature. It should be noted that even different instances of the same problem could be better optimized with different mutation operators. Among these methods are those that encourage exploration in the search space by mutating nearby solutions. An example of such an approach is the one referred to as "Neighborhood Mutation". The fundamental concept