Chapter 6
Hybridization of Cohort Intelligence and Fuzzy Logic (CIFL) for Truss Structure Problems

Saif Patel, Ishaan R. Kale, and Anand J. Kulkarni

Abstract Several nature-based optimization methods have been developed by the researchers to solve the real-world problems. There are certain characteristics of the inherent approaches associated with the algorithms which could be combined with other algorithm to enhance the exploration and exploitation quality of the algorithm. Cohort Intelligence (CI) is one of the socio-inspired optimization algorithms which is inspired from self-supervised learning of the candidates in a cohort. To further increase the performance of CI, it is hybridized with fuzzy logic (FL). FL is an approach that allows multiple possible truth values to be processed through variables. FL was used to solve problems with an open, imprecise data, and heuristics that make it possible to obtain accurate results. In this current work, a new combination of CI and FL named as CIFL is introduced for solving truss structure optimization problem. The validity of the algorithm is verified using two cases of three-bar truss design optimization problem. CIFL is applied to both discrete and continuous variable-constrained problems. The self-adaptive penalty function (SAPF) approach is used to handle the constraints. The results obtained from CIFL are compared with other nature-inspired optimization techniques and discussed in details.

Keywords Cohort intelligence · Self-adaptive penalty function · Fuzzy logic · Membership function · Truss structure

6.1 Introduction

Truss structure is one of the important real-world applications which typically used in bridges, towers, roofs, buildings, domes, various industrial sectors, etc. It is the most common lightweight structure used in practices. These systems are designed to meet

S. Patel · I. R. Kale (✉) · A. J. Kulkarni
Institute of Artificial Intelligence, Dr Vishwanath Karad MIT World Peace University, Pune 411038, India
e-mail: ishaan.kale@mitwpu.edu.in

A. J. Kulkarni
e-mail: anand.j.kulkarni@mitwpu.edu.in

I. R. Kale and A. Sadollah (eds.), *Optimization Methods for Structural Engineering*, Engineering Optimization: Methods and Applications,
https://doi.org/10.1007/978-981-99-2378-6_6

optimality criteria with the lowest weight, i.e., cost-effective with maximum reliability. These truss structures are associated with number of members which sustain load acting on the structure. In order to withstand for the long life, the mechanical conditions such as deflection of nodes and stress exerted in the member must be satisfied. In these truss structure problems, the number members are equal to number of design variables. The variables are of continuous and discrete types. Increasing the number of members, complexity of the problems increases which may not be handled using tradition gradient-based optimization techniques. There are several heuristic and metaheuristic techniques that have been proposed by the researchers and applied to solve these problems. Those are Bat Algorithm (BA) Yang and Gandomi (2012), Cuckoo Search Algorithm (CSA) (Gandomi et al. 2013), Mine Blast Algorithm (MBA) (Sadollah et al. 2013), Particle Swarm Optimization (PSO) (Li et al. 2009), Probability Collectives (PC) (Kulkarni et al. 2016), Cohort Intelligence (Kale and Kulkarni 2018, 2021). Furthermore, several hybrid optimization techniques have also been proposed and applied to solve these truss structure problems such as Particle Swarm Optimization and Genetic Algorithm (PSOGA) (Omidinasab and Goodarzimehr 2020), Cohort Intelligence with self-adaptive penalty function and Colliding Bodies Optimization (CI-SAPF-CBO) (Kale and Kulakrni 2021). In this work, the concept of fuzzy logic is incorporated in CI algorithm to solve the truss structure domain problems.

The concept of fuzzy is made from the things that are uncertain in nature. The theory of *absolute true* and *absolute false* doesn't exist in fuzzy logic. In last three decades, fuzzy set and fuzzy logic theory have been evolving and have been used in multiple engineering and natural socioeconomics sciences. Fuzzy logic model can replicate human way of thinking in complex situation and that's why it can be used as a tool examining natural complexity. Moreover, fuzzy logic can be exploited to predict chaotic behaviors. A fuzzy logic integrated with Genetic Programming (GP) is proposed by (Soh and Yang 2000) to increase the performance of the GP-based approach for structural optimization. Fuzzy set theory is employed to deal with imprecise and vague information, during structural design process. In Fuzzy Tuned Interactive Search Algorithm (FTISA) by (Mortazavi 2019) proposed mechanism evaluates agents via two predefined concepts named as Normalized Objective Function (NOF_i) and Normalized Members Density (NMD_i). The defined nine-rule fuzzy mechanism takes these values as input parameters and through fuzzification–defuzzification process returns a Topology/Size (TS) regulator value for each agent. The hybrid fuzzy genetic system for optimizing cabled-truss structures (Finotto et al. 2013) demonstrates an application of a hybrid fuzzy genetic system in the optimized lightweight structure is determined through a stochastic discrete topology and sizing optimization procedure that uses ground structure approach, nonlinear finite element analysis, genetic algorithm, and fuzzy logic. When performing optimization, the increase of the ground structure discretization led to a sharp increase of the search space. In addition, an increase in the number of evaluations of the FE model was also observed. This is because iterative procedures become part of the optimization problem when cable elements are used. For this reason, the effectiveness of the GA can be compromised since a relatively high number of evaluations may lead to a prohibitive computational cost.

In this work, an intrinsic property of cohort intelligence and fuzzy logic is combined together to design the new hybridized version named as CIFL. In the last two decades, fuzzy theory has been applied to structural optimization. Their results show that use of fuzzy set can mitigate the shortcomings of the aforementioned approaches, the current study deals with putting forward a new approach that takes into account the search space of size and topology optimization using fuzzy logic. This strategy does not only remove randomization but also decreases the convergence rate. The proposed CIFL is examined on two cases of three-bar truss structural test problems. These problems are having continuous as well as discrete variable with nonlinear constraints. To handle the constraints a self-adaptive penalty function (SAPF).

The chapter is organized as follows: Sect. 6.2 describes concept of CI algorithm. Section 2.1 explains the constrained handling SAPF approach. Section 6.3 describes detailed architecture of FL algorithm. The framework of hybrid CIFL is presented in Sect. 6.4. In Sect. 6.5, two cases of three-bar truss structure problems are presented. In Sect. 6.6, the results are analyzed and discussed with other contemporary algorithm. Section 6.7 concludes the proposed work.

6.2 Cohort Intelligence (CI) Algorithm

The CI method is inspired from social tendency of learning by cohort candidates through interaction and competition with every other candidate (Kulkarni et al. 2013). The cohort candidates with certain qualities make a particular behavior. Every candidate in a cohort follows certain behavior and adopts the associated qualities which assists to improve the behavior of individual candidates. This makes every candidate learn from one another and helps to evolve the overall cohort behavior. The cohort behavior is considered saturated if for considerable number of learning attempts (iterations) the behavior of the candidates does not improve and becomes almost same. The characteristics of CI algorithm (Kale and Kulkarni 2021) are as follows:

1. CI models the social behavior of learning candidates having common aim to achieve the best behavior by improving their individual qualities.
2. For every learning attempt, the cohort candidates are keen to improve its individual behavior by observing self and other candidate’ behavior in a cohort.
3. Every candidate of CI algorithm updates its search space for every learning attempt using sampling space reduction factor.
4. The CI algorithm has an ability to solve problems having more number variables and constraints.

6.2.1 Self-Adaptive Penalty Function (SAPF)

SAPF approach is adopted from Kale and Kulkarni (2021). The mechanism of SAPF is as follows:

The penalized or pseudo-objective function is written as

$$\phi\left(X^c\right) = f\left(X^c\right) + \mathrm{SAPF}\left(X^c\right),$$

where $\mathrm{SAPF}(X^c) = f(X^c) \times \left(\sum_{i=1}^{p} g_i(X^c) + \sum_{i=1}^{m} h_i(X^c)\right)$ is the self-adaptive penalty function;

$f(X^c)$ is the behavior of individual candidate.

The penalty parameter used in SAPF approach is itself a behavior of an individual candidate, i.e., $f(X^c)$. Every candidate has a different penalty parameter, and it updates iteratively as the CI algorithm progresses.

6.3 Fuzzy Logic

Fuzzy represents unclear/vague/absolute, e.g., ON–OFF, 0–1, High-Low, True–False, etc. However, there are several applications or the situations where these vague/absolute outputs would not work which requires the degree of truth. Lotfi Zadeh proposed the fuzzy logic (FL) in 1960 to represent the vague information (fuzzy sets) in the form of actual degree of truth (crisp value). FL is a concept associated with conventional logic which handles the information with partial truth (i.e., completely true or completely false). However, in the real-world applications or in day-to-day life activities it is necessary to analyze the exact degree of that partial truth. For that, FL method models the membership functions using certain rules based on the behavior of application. This rule-based system dives by an inference engine and provides the prescribed degree of truth. The FL methodology consists of four steps, such as (i) Fuzzification, (ii) Rule-base, (iii) Inference engine, and (iv) Defuzzification. The FL architecture is presented in Sect. 3.1.

6.3.1 Fuzzy Logic Architecture

The architecture of FL is divided into four parts as follows:

1. **Fuzzification**—It is process of converting the crisp value input (precise value) in fuzzy inputs using the membership function defined for that application. The membership function is considered for the optimization of truss structure best, mean, and worst function value.
2. **Rule Base**—In rule base, the defined *if–then* conditions are stored which is further used to control the decision-making system.

3. **Inference Engine**—The inference engine used to process fuzzy input. It evaluates the degree of match between fuzzy set and defined rules. Based on the percentage of degree of match, the rules are further modified and implemented to develop the control action.
4. **Defuzzification**—The processed fuzzy output generated by inference engine is then converted into crisp value using defuzzification step.

The general block diagram of FL architecture is presented in Fig. 6.1.

The term membership function used in fuzzification step specifies the degree of match of given input belonging to available sets. The degree of membership is represented between 0 and 1 which specifies the level of match of particular input belong to its set. This is also referred as the membership value of that variable. Different membership functions used to fuzzify a crisp (numerical) value are present in Fig. 6.2.

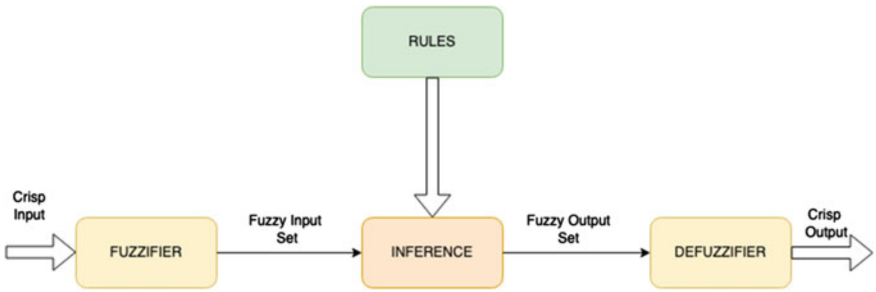

Fig. 6.1 Fuzzy logic architecture

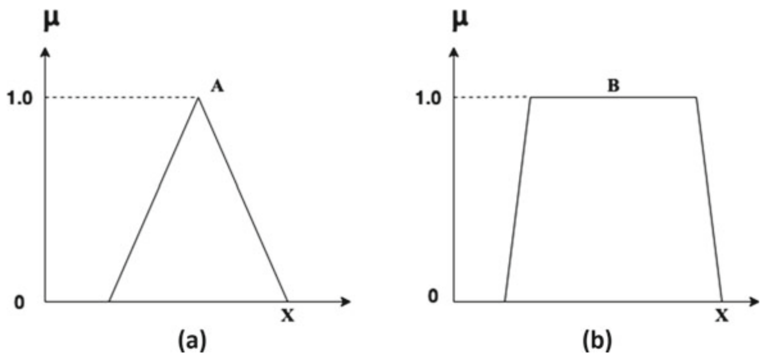

Fig. 6.2 Examples of triangular and trapezoidal membership function

6.4 Framework of CIFL

The proposed CIFL procedure is explained as follows:

Initialization of the number of cohort candidates C, variables t, sampling interval reduction factor R, and sampling interval Ψ.

$$\text{Minimize } f(x) = f(x_1, \ldots, x_i, \ldots, x_t) \tag{6.1}$$

$$\text{Subjected to } \Psi^{\text{lower}} \le x_i \le \Psi^{\text{upper}}, \quad i = 1, \ldots, t$$

Calculate probability of candidates based on objective function

$$p^c = \frac{{}^1\!/_{f(x^c)}}{\sum_{c=1}^{c} {}^1\!/_{f(x^c)}}, \quad (c = 1, \ldots, C) \tag{6.2}$$

By using roulette wheel approach, it can decide which candidate to follow. The candidate that is being followed has produced the best results. This approach gives the candidates a choice to follow better behavior that their existing one.

Every candidate C shrinks the sampling interval Ψ associated with each variable t based on whether condition is saturated or not. The cohort behavior is considered to be saturated if there are no further improvements, the results are observed.

After that every candidate forms a behavior by sampling qualities from within the updated sampling intervals. Then it uses the updated sampling intervals for the membership function. Use updated sampling intervals as crisp input and convert them into fuzzy set. Create fuzzy rule base for the algorithm. After that convert the fuzzy set into crisp output. Check if the obtained solution is converged; if not, start the process again from calculating the objective function. A FL hybridized with CI to increase the performance of the CI algorithm for structural optimization. Fuzzy set theory is employed to deal with imprecise and vague information, during structural design process.

In CI algorithm, the solutions are randomly generated using uniformly distributed approach within its sampling interval. This range is iteratively modified using a sampling space reduction factor R. In the current work, the hybridization of FL with CI algorithm is demonstrated and it is referred as CIFL. In CI algorithm, modified sampling intervals for every candidate c associated with each variable t. Whereas, in CIFL these modified sampling intervals are treated as crisp input set which further utilized in the fuzzification process. In fuzzification, the crisp input is converted into fuzzy set which is defined as *best*, *medium,* and *worst,* respectively. Further inference engine helps to determine the degree of match of fuzzy input using *if–then* rule. As the algorithm progresses, the degree of match iteratively updated based on the modified sampling intervals. Next step is to defuzzify the fuzzy input into crisp values using Eq. (6.3).

$$x^* = \frac{\int \mu_{\overline{c}}(x).x\mathrm{d}x}{\int \mu_{\overline{c}}(x).\mathrm{d}x} \tag{6.3}$$

After the defuzzification, these crisp values are further used to evaluate the function. If after significant number of iterations, the solution is converged and the solution does not improve and also becomes the same accept it as the final solution and stop, else repeat the process from the objective function. The flowchart of the proposed CIFL is presented in Fig. 6.3.

The hybrid CIFL algorithm was coded in Python3 on Visual Studio Code Platform with an Apple M1 chip @3.2 GHz octa-core processor with 8 GB RAM at Institute of Artificial Intelligence, MIT World Peace University, Pune, India (Fig. 6.4).

6.5 Three-Bar Truss Structure Problems

There are many heuristic as well as metaheuristic techniques have been used to solve the three-bar truss design optimization problem such as Swarm Optimization Approach (SOA) (Ray and Saini 2001), Cuckoo Search Algorithm (CSA) (Gandomi et al. 2013), Bat Algorithm (BA) (Yang and Gandomi 2012), Mine Blast Algorithm (MBA) (Sadollah et al. 2013), Cricket Algorithm (CA) (Canayaz and Karci 2016), Artificial Atom Algorithm (A^3) (Yildirim and Karci 2018).

Case 1: Three-bar truss structure is shown in Fig. 6.5. The volume of the truss structure is to be minimized subject to stress constraints.

There are two design variables (x_1, x_2) and three nonlinear constraint functions in this problem. The problem is expressed mathematically as follows:

Objective function:

$$\text{Min } f(x) = \left(2\sqrt{2}x_1 + x_2\right) \times L \tag{6.4}$$

Constraints:

$$g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0 \tag{6.5}$$

$$g_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \le 0 \tag{6.6}$$

$$g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \le 0 \tag{6.7}$$

where $0 \le x_1, x_2 \le 1$. The constants are $L = 100$ cm, $P = 2$ KN/cm^2, and $\sigma = 2$ KN/cm^2

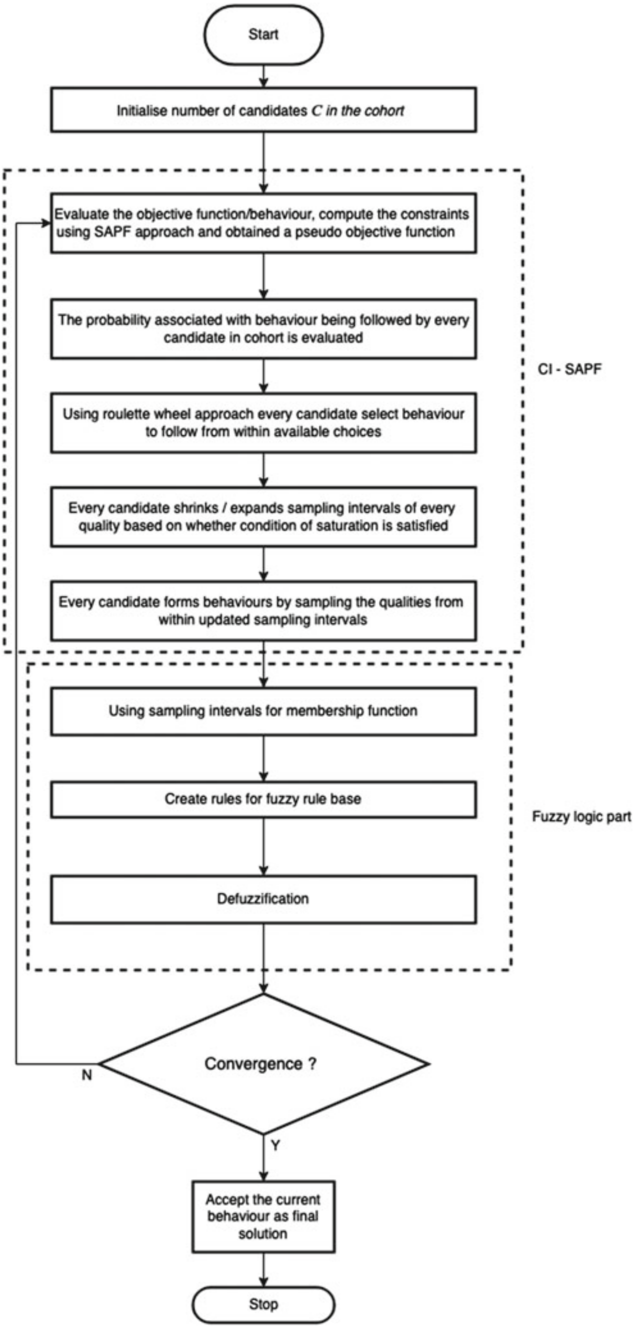

Fig. 6.3 Hybrid CIFL flowchart

C Number of cohort candidates ($c = 1, \ldots, C$)
Ψ Sampling interval
R Sampling interval reduction factor
P Penalty function
g Constraint violations

Initialize C, Ψ, R
For
1. The set of design variables ($\boldsymbol{X}$) are initialized using uniform distribution method within the defined sampling space Ψ.
2. Evaluate objective function value $f(X^c)$
3. SAPF approach applied to handle the constraints and generate a pseudo behavior:
   $\phi(X^c) = f(X^c) + P(X^c)$
   where $P(X^c) = f(X^c) \times \sum_{C=1}^{C} g\,(X^c)$
4. Calculate the probability $p^c$of every candidate behavior: $p^c = \frac{1/\phi^*(X^c)}{\sum_{C=1}^{C} 1/\phi^*(X^c)}$
5. Roulette wheel approach is used for every candidate c to select and follow a behavior from within C available choices.
6. Every candidate c updates its sampling interval $\boldsymbol{\Psi}^c$ using sampling space reduction parameter R and $\boldsymbol{X}^c$:
   $\boldsymbol{\Psi}^c = [\boldsymbol{\Psi}^{c,lower}, \boldsymbol{\Psi}^{c,upper}] = [\boldsymbol{X}^c - \left\|\frac{\boldsymbol{\psi}^{upper} - \boldsymbol{\psi}^{lower}}{2}\right\| \times R, \boldsymbol{X}^c + \left\|\frac{\boldsymbol{\psi}^{upper} - \boldsymbol{\psi}^{lower}}{2}\right\| \times R$
7. The updated sampling intervals $\boldsymbol{\Psi}^c$ are used as a range for membership function for every candidate.
8. Create *if-then* rules for fuzzy rule base.
9. Defuzzification: Centroid method: $x^* = \frac{\int \mu_{\tilde{c}}(x).\ x\,dx}{\int \mu_{\tilde{c}}(x).\ dx}$
10. Obtained solution
11. **If**: The solution $\phi(X^c)$ saturates and no further improvement in it.
    Update the sampling space is set $\boldsymbol{\Psi}^c$ as original $\boldsymbol{\Psi}$
    Accept the solutions.
    Else:
    Continue to step 2

Fig. 6.4 Hybrid CIFL pseudo-code

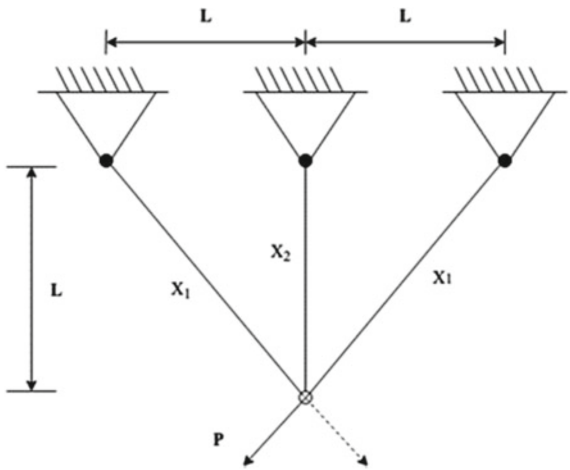

Fig. 6.5 Three-bar truss design Case 1 (Ayse and Karci 2018)

Case 2: The indeterminate three-bar truss structure (refer Fig. 6.6) is subject to vertical and horizontal forces at a single node which is an intersection of all the three members. The aim is to minimize the structural weight W and is minimized under the constraint that the stress in all members should be smaller than allowable stress σ_0 in absolute magnitude. After nondimensionalization of the objective function and variables, $F = \sigma_0 W/Ppl$ and $x_i = a_i\sigma_0/P$. This problem is previously

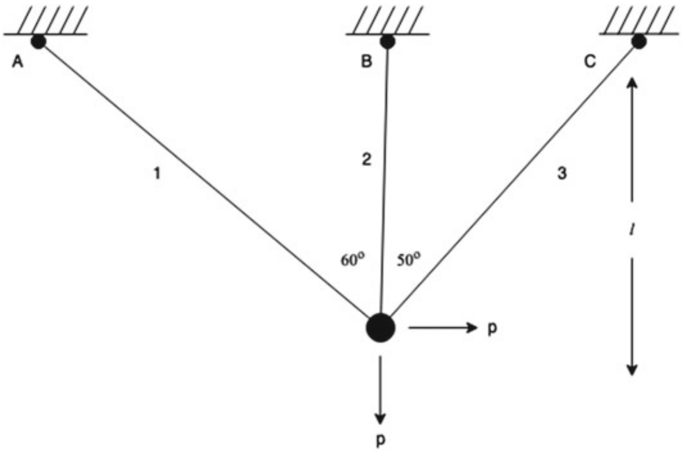

Fig. 6.6 Three-bar truss design Case 2 (Shin et al. 1990)

solved by using CI-SPF (Kale and Kulkarni 2018), Multi-Random Start Local Search (MRSLS), CI-SAPF, and CI-SAPF-CBO (Kale and Kulkarni 2021).

$$\text{Objective function Min: } f(x) = 2x_1 + x_2 + \sqrt{2x_3} \tag{6.8}$$

Subject to:

$$g_1 = 1 - \frac{\sqrt{3}x_3 + 1.932x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.9}$$

$$g_2 = 1 - \frac{0.634x_1 + 2.828x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.10}$$

$$g_3 = 1 - \frac{0.5x_1 + 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.11}$$

$$g_4 = 1 + \frac{0.5x_1 - 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_3} \geq 0 \tag{6.12}$$

$$x_i = \{0.1,\ 0.2,\ 0.3,\ 0.5,\ 0.8,\ 1.0,\ 1.2\},\quad i = 1, 2, 3$$

6.6 Results Analysis and Discussion

The algorithm is validated by solving two cases of three-bar truss structure-constrained optimization problems for the minimization of weight. The solutions obtained from proposed CIFL algorithm for three-bar truss structure Case 1 problem are compared with Swarm Optimization Algorithm (SOA), Cuckoo Search Algorithm (CSA), Bat Algorithm (BA), Mine Blast Algorithm (MBA), Cricket Algorithm (CA), and Artificial Atom Algorithm (A^3) presented in Table 6.1. The statistical results such as best, mean, and worst function values, standard deviation, average CPU time and average function evaluations are obtained from 30 independent trails. From the result comparison, it shows that the proposed CIFL algorithm yielded satisfactory results compared with other algorithms (refer Table 6.1). It is observed that the solution precisely worst by 0.19% as compared to the latest solution obtained by A^3 (Yildirim and Karci 2018) and CA (Canayaz and Karci 2016). The convergence plot of Case 1 of three-bar truss structure problem is presented in Fig. 6.7.

In CSA (Gandomi et al. 2013), Lévy's flight approach was employed with three key rules such as selection of the best, exploitation by local random walk, and exploration by randomization. The performance of the CSA is dependent on the parameter which needs to be fine-tuned. It may require certain preliminary trail to set an appropriate value. The CSA adopts Lévy's flight strategy so that only best solution can be obtained which is close to optimal value. Like CSA, Bat Algorithm (BA) (Yang and Gandomi 2012) is also required fine-tuning of two computational parameters which directly affect the convergence of BA. The Bat Algorithm (BA) models the foraging behavior of bats. Bat uses echolocation to sense the distance food and background barrier. The Cricket Algorithm (CA) (Canayaz and Karci 2016) which models the behavior of cricket insect. These cricket insects intercommunicate with their peers through the sound in nature. They generate the sound by chirping of their wings based on the atmospheric temperature. An A^3 (Yildirim and Karci 2018) referred to

Table 6.1 Comparison of results for three-bar truss structure optimization problem Case 1

Algorithm	x_1	x_2	$f(x)$
SOA (Ray and Saini 2001)	0.79500	0.39500	264.3000
CSA (Gandomi et al. 2013)	0.78867	0.40902	263.9716
BA (Yang and Gandomi 2012)	0.78863	0.43838	263.8962
MBA (Sadollah et al. 2013)	0.7885650	0.4082482	263.8958
CA (Canayaz and Karci 2016)	0.78863	0.408368	263.8958
A^3 (Yildirim and Karci 2018)	0.7887357	0.408078	263.8958
CIFL	0.7049	0.5800	264.4031

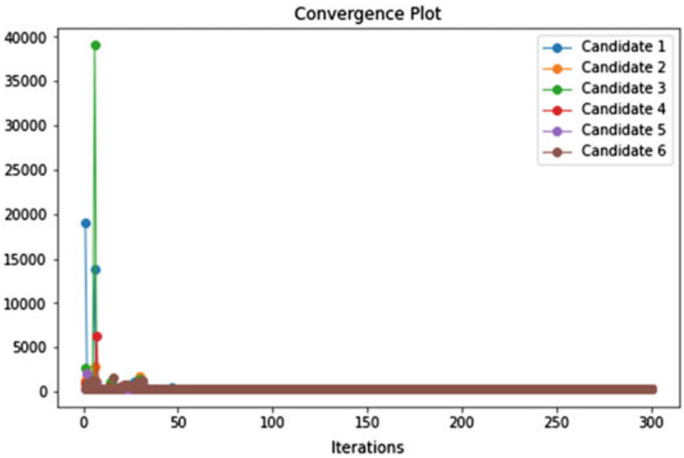

Fig. 6.7 Conversion plot for three-bar truss structure optimization problem Case 1

as Artificial Atom Algorithm is based on chemical compounding processes. There are two operators in A^3 which are iconic bond and covalent bond. It is associated with the conceptual strategy of an electron, atom, and atom set. Number of design variables are considered while determining number of electrons. The atom is formed randomly according to constraint conditions (Table 6.2).

For three-bar Case 2 truss structure problem, CIFL is compared with CI-SPF, CI-SAPF, MRSLS, CI-SAPF-CBO algorithm presented in Table 6.3. The statistical results obtained from 30 trials are presented in Table 6.4. It is observed that the CIFL algorithm has obtained same function value. The convergence plot of Case 2 of three-bar truss structure problem is presented in Fig. 6.8. The average computational time obtained from CIFL for Case 1 and Case 2 is 7.04 and 7.55 s, respectively. The constraints associated with these problems are very much crucial task. Here, in CIFL algorithm SAPF approach is incorporated to deal with the constraints. This is important to note that it does not need to set a penalty parameter. It is adaptively set as the algorithm progresses. This SAPF approach is also used in CI-SAPF, CI-SAPF-CBO, and MRSLS. In CI-SPF, the static penalty function is incorporated to deal with the constraints; however, it requires to fine-tuning to set an appropriate value of penalty parameter.

Table 6.2 Statistical results obtained from CIFL for three-bar truss design optimization problem Case 1

Best	264.4031
Mean	273.9651
Worst	286.3454
Standard deviation	6.249121
Avg. function evaluations	821
Avg. CPU time (s)	7.04

Table 6.3 Comparison of results for three-bar truss design optimization problem Case 2

Algorithms	Function value	Optimum variables
NEWSUMT-A (Shin et al. 1990)	3.0414	[1.2, 0.5, 0.1]
CI-SPF (Kale and Kulkarni 2018)	3.0414	[1.2, 0.5, 0.1]
MRSLS (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CI-SAPF (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CI-SAPF-CBO (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CBO (Kale and Kulkarni 2021)	3.0414	[1.2, 0.5, 0.1]
CIFL	3.0414	[1.2, 0.5, 0.1]

Table 6.4 Statistical results for three-bar truss design optimization problem Case 2

Best	3.0414
Mean	3.5050
Worst	3.7071
Standard deviation	0.1511
Avg. function evaluation	250
Avg. CPU time (s)	7.55

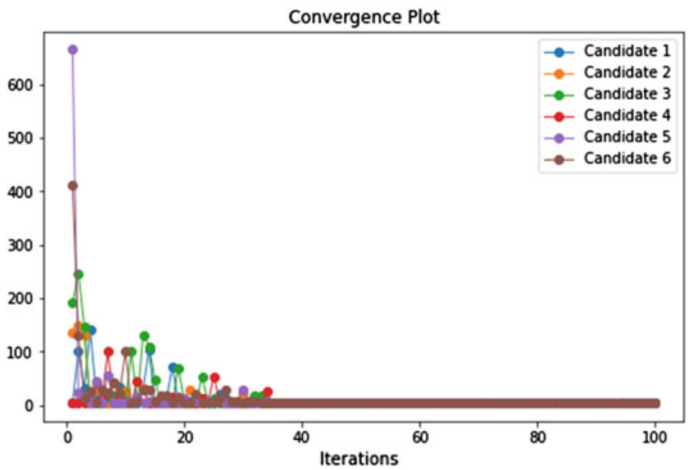

Fig. 6.8 Conversion plot for three-bar truss design optimization problem Case 2

In CIFL as well every candidate updates its sampling interval for every iteration using a sampling space reduction factor. It also requires certain preliminary trials however, according to the analysis conducted in Kale and Kulkarni (2018) the value of R can be set between 0.95 and 0.97. This eliminates the tuning of parameter R. The role of FL in CIFL is to nullify the randomly regeneration of variable values. The sampling space updates in every iteration are considered for the membership function. Then using a rule base condition, the fuzzy system provides the crisp values which are further used as a set of variables.

6.7 Conclusions

The CIFL algorithm is successfully applied to solve the two cases of constrained three-bar truss structure problem. These problems are associated with discrete as well as continuous variables and have nonlinear constraints. For discrete variables, a round-off integer sampling is incorporated. The performance of CIFL is observed to be precisely similar as compared to other metaheuristic algorithms. The hybrid version of CIFL algorithm eliminates the randomly generated solutions which in CI algorithm for every learning attempt. The SAPF approach is incorporated to handle the constraints associated with the problems. After the extensive comparative study, it is observed that CIFL obtained satisfactory results as compared to other contemporary metaheuristic algorithms. The CIFL algorithm can be used to solve more design engineering problems as well as structural engineering problems.

References

Canayaz M, Karci A (2016) Cricket behaviour-based evolutionary computation technique in solving engineering optimization problems. Appl Intell 44(2):362–376

Finotto VC, da Silva WR, Valášek M, Štemberk P (2013) Hybrid fuzzy-genetic system for optimising cabled-truss structures. Adv Eng Softw 62:85–96

Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35

Kale IR, Kulkarni AJ (2021) Cohort intelligence with self-adaptive penalty function approach hybridized with colliding bodies optimization algorithm for discrete and mixed variable constrained problems. Complex Intell Syst 7(3):1565–1596

Kale IR, Kulkarni AJ (2018) Cohort intelligence algorithm for discrete and mixed variable engineering problems. Int J Parallel Emergent Distrib Syst 33(6):627–662

Li LJ, Huang ZB, Liu F (2009) A heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 87(7–8):435–443

Mortazavi A (2019) Interactive fuzzy search algorithm: a new self-adaptive hybrid optimization algorithm. Eng Appl Artif Intell 81:270–282

Omidinasab F, Goodarzimehr V (2020) A hybrid particle swarm optimization and genetic algorithm for truss structures with discrete variables. J Appl Comput Mech 6(3):593–604

Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng Optim 33(6):735–748

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612
Shin DK, Gurdal Z, Griffin OH (1990) A penalty approach for nonlinear optimization with discrete design variables. Eng Optim 16(1):29–42
Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput
Yildirim AE, Karci A (2018) Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm. In: 2018 international conference on artificial intelligence and data processing (IDAP), pp 1–5
Yang Y, Soh CK (2000) Fuzzy logic integrated genetic programming for optimization and design. J Comput Civ Eng 14(4):249–254