# Constrained Optimization Based on Hybridized Version of Superiority of Feasibility Solution Strategy

Wali Khan Mashwani<sup>1</sup>, Faiz Ur Rehaman<sup>1</sup>, Habib Shah<sup>2</sup>

<sup>a</sup>Institute of Numerical Sciences, Kohat University of Science & Technology, Pakistan Email:mashwanigr8@gmail.com <sup>b</sup>College of Computer Science, King Khalid University,Abha, Saudi Arbia,Email:

habibshah.uthm@gmail.com

#### Abstract

Teaching learning based optimization (TLBO) is a stochastic algorithm which was first proposed for unconstrained optimization problems. It is population based, nature-inspired, and meta-heuristic that imitates teaching learning process. It has two phases, teacher and learner. In teacher phase, the teacher who is well-learned person transfers his/her knowledge to the learners to raise their grades/results; while in learner phase, learners/pupils learn and refine their knowledge through mutual interconnection. To solve constrained optimization problems (COPs) through TLBO we need to merge it with some constraint handling technique (CHT). Superiority of feasibility (SF) is a concept for making CHTs, existed in different forms based on various decisive factors. Most commonly used decision making factors in SF are number of constraints violated (NCV) and weighted mean (WM) values for comparing solutions. In this work, SF based on number of constraints violated (NCVSF) and weighted mean (WMSF) are incorporated in the framework of TLBO. These are tested upon CEC-2006 constrained suit with the remark that single factor used for the decision making of winner is not a wise idea. Mentioned remark leads us to made a single CHT that carries the capabilities of both discussed CHTs. It laid the foundation of hybrid superiority of feasiblity (HSF); where NCV and WM factors are combined with giving dominance to NCV over WM. In current research three constrained versions of TLBO are formulated by the name NCVSF-TLBO, WMSF-TLBO, and HSF-TLBO; while implanting NCVSF, WMSF, and HSF in the framework of TLBO, respectively. These constrained versions of TLBO are evaluated on CEC-2006 with the remarks that HSF-TLBO got prominent and flourishing status among these.

*Keywords:* Optimization, Constrained Optimization, Evolutionary Algorithm, Teaching learning based optimization.

Preprint submitted to Elsevier

 $<sup>^{\</sup>star} \mathrm{Constrained}$  Optimization Based on Hybridized Version of Superiority of Feasibility Solution Strategy

<sup>\*</sup>Corresponding author

Email address: mashwanigr80gmail.com (Habib Shah<sup>2</sup>)

#### 1. Introduction

Optimization is the mathematical process to find the best possible solution vector of decision variables that minimize or maximize the objective function of the given optimization and search problems. Optimization is natural process such as maximize the profit and minimize expenditure in daily life applications. Optimization has become an essential part in all branches of sciences and engineering technologies since its first piratical application [1, 2]. The standard form of constrained optimization problem is described as follow:

$$Minimizef(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix}$$
(1)

subject to the following conditions:

$$g_{i}(x) \leq 0, \quad i = 1, 2, \dots, p$$

$$h_{j}(x) = 0, \quad j = 1, 2, \dots, q$$

$$x_{l}^{i} \leq x^{i} \leq x_{u}^{i}, \quad i = 1, 2, \dots, N$$

$$(2)$$

where  $x = (x_1, x_2, \ldots, x_n)^T \in \Omega$  is a vector of *n* decision variables, f(x) consist of *m* objective functions,  $g_i(x)$  describes *P* inequality functions and  $h_j(x)$  denotes *q* equality functions. If  $\Omega$  is a closed and connected region in  $\mathbb{R}^n$  and all objective functions are described in real valued variables, then problem (1) is said to be continuous multi-objective optimization problem (MOP).

In global optimization, the main objective is to find the best solution for the given optimization problem in presence of the multiple optimal solutions. In constrained optimization, one has to find the feasible solution subject to several constraints functions [3, 4, 5].

In the last decade or two, evolutionary optimization algorithms have contributed much for dealing with different optimization and search problems. Unlike the local techniques, where a single design point is updated, evolutionary algorithms do not require any gradient information and typically utilize a set of solutions to find optimum solution for the given problem [6, 4, 7, 8, 9, 10]. This family of algorithms are typically inspired from the phenomena of nature. EAs have the many advantages including extremely robust and easy to implement and being well suited for discrete optimization problems [11, 12, 13]. The classical evolutionary algorithms can be distinguished by the nature of their solutions representation and operators employing for their evolution. The evolutionary strategies employs mutation operator to create new solutions which are represented in real-numbers [14, 15, 16], evolutionary programming also required solutions that are represented in real numbers or integers [17] and genetic algorithms employ crossover operators to evolve its population [18, 19] and genetic programming required tree based representation of computer programs to perform search process [20, 21, 22]. The main drawbacks associated with most of the classical EAs are their high computational cost, poor constraint-handling abilities, problem-specific parameter tuning and limited problem size and lack of ability to cope with large scale global optimization problems.

Teaching learning based optimization (TLBO) is one of the most efficient and recently developed population based EA. TLBO was proposed by R.Venkata Rao and his colleagues [23, 24]. TLBO is a stochastic algorithm and was first applied on unconstrained optimization problems. It is nature-inspired populationbased algorithm that imitates teaching learning process. It has two phases, teacher and learner. In teacher phase, the teacher who is well-learned person transfers his/her knowledge to the learners to raise their grades/results; while in learner phase, learners/pupils learn and refine their knowledge through mutual interconnection It is the most popular algorithms in the field of optimization. It is used on large scale in various fields of engineering and industry. In this algorithms the best solution play the role of a teacher. This algorithms is different from EAs because here the member of swarm improves their results/grades in a given search space. In TLBO, the starting population is randomly selected in the given search space. The members of the swarm updates his position as given below[23]: New positions are updated as follows, in teacher phase of TLBO.

$$\mathbf{x}_{i}^{t+1} = \mathbf{x}_{i}^{t} + r \times (\mathbf{x}_{best}^{t} - \mathbf{F} \times \boldsymbol{\mu}^{t}), \tag{3}$$

where  $\mathbf{x}_i^{t+1}$  is the updated position, r is randomly generated numbers between 0 and 1,  $\mathbf{x}_i^t$  is the current position,  $\mathbf{x}_{best}^t$  is the best position,  $\mathbf{F}$  is randomly generated teaching factor either 1 or 2,  $\mu^t$  is mean position of the population in iteration t. In teacher phase of TLBO positions are updated according to Eq.(3).

$$\mathbf{x}_{i}^{t+1} = \begin{cases} \mathbf{x}_{i}^{t} + r \times (\mathbf{x}_{j}^{t} - \mathbf{x}_{i}^{t}), & \text{if } f(\mathbf{x}_{i}^{t}) > f(\mathbf{x}_{j}^{t}); \\ \mathbf{x}_{i}^{t} + r \times (\mathbf{x}_{i}^{t} - \mathbf{x}_{j}^{t}), & \text{else.} \end{cases}$$
(4)

where  $\mathbf{x}_i^{t+1}$  is the updated position, r is randomly generated numbers between 0 and 1,  $\mathbf{x}_i^t$  is the current position,  $\mathbf{x}_j^t$  is the position of randomly selected solution from the population other than  $i^{th}$  one,  $f(\mathbf{x}_i^t)$  and  $f(\mathbf{x}_j^t)$  are the fitness values of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  respectively. In learner phase of TLBO positions have been updated according to Eq.(4).

### 1.1. Number of Constraints Violated based SF (NCVSF)

In NCVSF the decisive factor for declaring the winner among the solution is number of constraints violated (NCV). According to it the comparison of two solutions  $\mathbf{x}_{i}^{t}$  and  $\mathbf{x}_{i}^{t}$  are done as follow:

$$\mathbf{x}_{i}^{t+1} = \begin{cases} \text{Winner by } NCV \text{ values,} & \text{if } NCV(\mathbf{x}_{i}^{t}) \neq NCV(\mathbf{x}_{j}^{t}); \\ \text{Winner by cost values,} & \text{else,} \end{cases}$$
(5)

where  $\mathbf{x}_i^{t+1}$  is updated position,  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  are competing positions, and NCV is number of constraints violated. Pseudo-code of NCVSF technique is given in Algorithm 1:

Algorithm 1 Pseudo-code of NCVSF

Comparing positions  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  via NCVSF;  $NCV(\mathbf{x}_i^t) =$ Number of constraints violated by  $\mathbf{x}_i^t$ ;  $f(\mathbf{x}_i^t) =$ Fitness value of  $\mathbf{x}_i^t$ ;  $NCV(\mathbf{x}_j^t) =$ Number of constraints violated by  $\mathbf{x}_j^t$ ;  $f(\mathbf{x}_j^t) =$ Fitness value of  $\mathbf{x}_j^t$ ; **if**  $NCV(\mathbf{x}_i^t) \neq NCV(\mathbf{x}_j^t)$  **then** Winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected by their NCV values; **else** Winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected by their cost values; **end if** 

### 1.2. Weighted Mean based Superiority of Feasibilities (WMSF)

In WMSF, the factor upon which the superior among the compared solution are nominated is weighted mean. In this CHT winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected as follow:

$$\mathbf{x}_{i}^{t+1} = \begin{cases} \text{Winner by } WM \text{ values, } \text{if } WM(\mathbf{x}_{i}^{t}) \neq WM(\mathbf{x}_{j}^{t}); \\ \text{Winner by cost values, } \text{else,} \end{cases}$$
(6)

where  $\mathbf{x}_i^{t+1}$  is updated position,  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  are competing positions, and WM is weighted mean of all constraints violations which is defined as follow [25].

$$\overline{v}(\mathbf{x}) = \frac{\sum_{i=1}^{n} w_i(\max(0, G_i(\mathbf{x})))}{\sum_{i=1}^{n} w_i}$$
(7)

where

$$G_i(\mathbf{x}) = \begin{cases} g_j(\mathbf{x}) & j = 1, \dots, l \\ |h_k(\mathbf{x})| - \epsilon & k = l, \dots, n. \end{cases}$$
(8)

 $w_i(=1/G_{max_i})$  is a weight parameter,  $G_{max_i}$  is the maximum violation of constraint in combined population. Pseudo-code of WMSF technique is given in Algorithm 2:

### Algorithm 2 Pseudo-code of WMSF

Comparing positions  $\mathbf{x}_{i}^{t}$  and  $\mathbf{x}_{j}^{t}$  via WMSF;  $\overline{v}(\mathbf{x}_{i}^{t}) =$  Weighted mean of constraints violations of  $\mathbf{x}_{i}^{t}$ ;  $f(\mathbf{x}_{i}^{t}) =$  Fitness value of  $\mathbf{x}_{i}^{t}$ ;  $\overline{v}(\mathbf{x}_{j}^{t}) =$  Weighted mean of constraints violations of  $\mathbf{x}_{j}^{t}$ ;  $f(\mathbf{x}_{j}^{t}) =$  Fitness value of  $\mathbf{x}_{j}^{t}$ ; **if**  $\overline{v}(\mathbf{x}_{i}^{t}) \neq \overline{v}(\mathbf{x}_{j}^{t})$  **then** Winner of  $\mathbf{x}_{i}^{t}$  and  $\mathbf{x}_{j}^{t}$  is selected by their WM values; **else** Winner of  $\mathbf{x}_{i}^{t}$  and  $\mathbf{x}_{j}^{t}$  is selected by their cost values; **end if** 

#### 2. Main Research Contributions

Using single decisive factors NCV or WM solely for SF reduces the diversity among the solutions; while pushing them in the specific direction of used criteria. In NCVSF when we meet up with a situation when both competing solutions are infeasible with same NCV values the winner is decided based on fitness values which is not a wise move. Similar situation is happened in WMSF when the WM values for two infeasible solution are same the winner is decided upon fitness values, again it is not considered a good decision. In this work discussed gaps are tried to remove by introducing HSF. To take step based on fitness values creates infeasibility among solutions; while neglecting the constraints violations. This gap is filled up by taking step via discussed CHT in each proposed algorithm. Details of the prescribed contributions are given in below.

### 2.1. Hybrid Superiority of Feasibilities (HSF)

HSF is hybrid of NCVSF and WMSF. In it the decisive factor NCV is considered superior over weighted mean value. It is due to giving more impotence to NCV since it is considered superior than WM values. Here winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected as follow:

$$\mathbf{x}_{i}^{t+1} = \begin{cases} \text{Winner by } NCV \text{ values,} & \text{if } NCV(\mathbf{x}_{i}^{t}) \neq NCV(\mathbf{x}_{j}^{t}); \\ \text{Winner by } WM \text{ values,} & \text{if } NCV(\mathbf{x}_{i}^{t}) = NCV(\mathbf{x}_{j}^{t}) \text{ and } WM(\mathbf{x}_{i}^{t}) \neq WM(\mathbf{x}_{j}^{t}); \\ \text{Winner by cost values,} & \text{else,} \end{cases}$$
(9)

where  $\mathbf{x}_i^{t+1}$  is updated position,  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  are competing positions, NCV is number of constraints violated, and WM is weighted mean of all constraints violations. Pseudo-code of HSF technique is given in Algorithm 3:

### Algorithm 3 Pseudo-code of HSF

Comparing positions  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  via HSF;  $NCV(\mathbf{x}_i^t) =$  Number of constraints violated by  $\mathbf{x}_i^t$ ;  $\overline{v}(\mathbf{x}_i^t) =$  Weighted mean of constraints violations of  $\mathbf{x}_i^t$ ;  $f(\mathbf{x}_i^t) =$  Fitness value of  $\mathbf{x}_i^t$ ;  $NCV(\mathbf{x}_j^t) =$  Number of constraints violated by  $\mathbf{x}_j^t$ ;  $\overline{v}(\mathbf{x}_j^t) =$  Weighted mean of constraints violations of  $\mathbf{x}_j^t$ ;  $\overline{v}(\mathbf{x}_j^t) =$  Fitness value of  $\mathbf{x}_j^t$ ; if  $NCV(\mathbf{x}_i^t) \neq NCV(\mathbf{x}_j^t)$  then Winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected by their NCV values; else if  $NCV(\mathbf{x}_i^t) = NCV(\mathbf{x}_j^t)$  and  $\overline{v}(\mathbf{x}_i^t) \neq \overline{v}(\mathbf{y}_i^t)$  then Winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected by their WM values; else Winner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected by their WM values; else Minner of  $\mathbf{x}_i^t$  and  $\mathbf{x}_j^t$  is selected by their WM values; else

For constrained optimization taking step in the learner phase of TLBO according to fitness values is not a good idea, since solutions are neglecting constraints in each iteration that causes bringing in infeasibility among solutions; therefore, for constrained optimization selecting step via some CHT is good concept. Implantation of the mentioned concept is described in following:

1. Learner Phase based on NCVSF:

$$\mathbf{y}_{i}^{t} = \begin{cases} \mathbf{x}_{i}^{t} + r * (\mathbf{x}_{j}^{t} - \mathbf{x}_{i}^{t}) & \text{if winner of } \mathbf{x}_{i}^{t} \text{ and } \mathbf{x}_{j}^{t} \text{ is } \mathbf{x}_{j}^{t} \text{ based on NCVSF;} \\ \mathbf{x}_{i}^{t} + r * (\mathbf{x}_{i}^{t} - \mathbf{x}_{j}^{t}) & \text{else.} \end{cases}$$
(10)

2. Learner Phase based on WMSF:

$$\mathbf{y}_{i}^{t} = \begin{cases} \mathbf{x}_{i}^{t} + r * (\mathbf{x}_{j}^{t} - \mathbf{x}_{i}^{t}) & \text{if winner of } \mathbf{x}_{i}^{t} \text{ and } \mathbf{x}_{j}^{t} \text{ is } \mathbf{x}_{j}^{t} \text{ based on WMSF;} \\ \mathbf{x}_{i}^{t} + r * (\mathbf{x}_{i}^{t} - \mathbf{x}_{j}^{t}) & \text{else.} \end{cases}$$
(11)

3. Learner Phase based on HSF:

$$\mathbf{y}_{i}^{t} = \begin{cases} \mathbf{x}_{i}^{t} + r * (\mathbf{x}_{j}^{t} - \mathbf{x}_{i}^{t}) & \text{if winner of } \mathbf{x}_{i}^{t} \text{ and } \mathbf{x}_{j}^{t} \text{ is } \mathbf{x}_{j}^{t} \text{ based on HSF;} \\ \mathbf{x}_{i}^{t} + r * (\mathbf{x}_{i}^{t} - \mathbf{x}_{j}^{t}) & \text{else.} \end{cases}$$
(12)

In this paper, TLBO will be combine with NCVSF, WMSF, and HSF for handling the constraints of the problem (1). This will result in a new constrained version of TLBO, denoted by NCVSF-TLBO, WMSF-TLBO, HSF-TLBO. The pseudo code of these are given below.

### Algorithm 4 Pseudo code of NCVSF-TLBO

- 1: Set TLBO parameters: Pop size = nPop; Maximum iterations = MaxIt;
- 2: Stochastically Generate initial pop;
- 3: Set iteration counter t=0;
- 4: Evaluate costs and constrained violations for the pop;
- 5: Nominate BestSol= $\mathbf{x}^*$  using algorithm 1;
- 6: while  $t \leq MaxIt$  do
- 7: Set iteration counter t = t+1;
- 8: Find new positions via Eq.(3) and update positions via Algo. (1);
- 9: Find new positions via Eq.(10) and update positions via Algo. (1);
- 10: Update BestSol= $\mathbf{x}^*$  using algorithm 1;

11: end while

# Algorithm 5 Pseudo code of WMSF-TLBO

- 1: Set TLBO parameters: Pop size = nPop; Maximum iterations = MaxIt;
- 2: Stochastically Generate initial pop;
- 3: Set iteration counter t=0;
- 4: Evaluate costs and constrained violations for the pop;
- 5: Nominate BestSol= $\mathbf{x}^*$  using algorithm 2;
- 6: while  $t \leq MaxIt$  do
- 7: Set iteration counter t = t+1;
- 8: Find new positions via Eq.(3) and update positions via Algo. (2);
- 9: Find new positions via Eq.(11) and update positions via Algo. (2);
- 10: Update BestSol= $\mathbf{x}^*$  using algorithm 2;
- 11: end while

# Algorithm 6 Pseudo code of HSF-TLBO

- 1: Set TLBO parameters: Pop size = nPop; Maximum iterations = MaxIt;
- 2: Stochastically Generate initial pop;
- 3: Set iteration counter t=0;
- 4: Evaluate costs and constrained violations for the pop;
- 5: Nominate BestSol= $\mathbf{x}^*$  using algorithm 3;
- 6: while  $t \leq MaxIt$  do
- 7: Set iteration counter t = t+1;
- 8: Find new positions via Eq.(3) and update positions via Algo. (3);
- 9: Find new positions via Eq.(12) and update positions via Algo. (3);
- 10: Update BestSol= $\mathbf{x}^*$  using algorithm 3;
- 11: end while



Figure 1: Flow Chart of the Proposed HSF-TLBO

### 3. Simulation Results, Comparison, and Discussion

The basic properties of these problems are given in table 1, where n is the problem dimension,  $f(\mathbf{x})$  is know optimal solution at  $\mathbf{x}$ ,  $\rho$  denotes the ratio between feasible and whole search space, **LI** denote number linear inequality constraint, **NI** denotes the number of non-linear inequality constraints, **LE** denotes the number of linear equality constraints, **NE** denotes the number of non-linear equality constraints, **TC** denotes total constraints and a denotes active constraints at  $\mathbf{x}$ .

Prob.	n	$f(\mathbf{x})$	Type of function	ρ	LI	NI	LE	NE	TC	a
g01	13	-15.0000000000	quadratic	0.0111%	9	0	0	0	9	6
g02	20	-0.8036191042	non linear	99.9971%	0	2	0	0	2	1
g03	10	-1.0005001000	polynomial	0.0000%	0	0	0	1	1	1
g04	5	-30665.5386717834	quadratic	52.1230%	0	6	0	0	6	2
g05	4	5126.4967140071	cubic	0.0000%	2	0	0	3	5	3
g06	2	-6961.8138755802	cubic	0.0066%	0	2	0	0	2	2
g07	10	24.3062090681	quadratic	0.0003%	3	5	0	0	8	6
g08	2	-0.0958250415	non linear	0.8560%	0	2	0	0	2	0
g09	7	680.6300573745	polynomial	0.5121%	0	4	0	0	4	0
g10	8	7049.2480205286	linear	0.0010%	3	3	0	0	6	6
g11	2	0.7499000000	quadratic	0.0000%	0	0	0	1	1	1
g12	3	-1.0000000000	quadratic	4.7713%	0	1	0	0	1	0
g13	5	0.0539415140	non linear	0.0000%	0	0	0	3	3	3
g14	10	-47.7648884595	non linear	0.0000%	0	0	3	0	3	3
g15	3	961.7150222899	quadratic	0.0000%	0	0	1	1	2	2
g16	5	-1.9051552586	non linear	0.0204%	4	34	0	0	38	4
g17	6	8853.5396748064	non linear	0.0000%	0	0	0	4	4	4
g18	9	-0.8660254038	quadratic	0.0000%	0	13	0	0	13	6
g19	15	32.6555929502	non linear	33.4761%	0	5	0	0	5	0
g20	24	0.2049794002	linear	0.0000%	0	6	2	12	20	16
g21	7	193.7245100700	linear	0.0000%	0	1	0	5	6	6
g22	22	236.4309755040	linear	0.0000%	0	1	8	11	20	19
g23	9	-400.0551000000	linear	0.0000%	0	2	3	1	6	6
g24	2	-5.5080132716	linear	79.6556%	0	2	0	0	2	2

Table 1: Basic Properties of the Benchmark Functions

The CEC 2006 constrained optimization problems suit [26] containing of 24 problems has been selected from available literature, to test the robustness and consistency of the proposed algorithms. The results are compiled for all problems in 25 independent run. For commencement of simulations, the parameters are set for selected schemes as follow.

### 3.1. PC and software Specification

The system which is used for performing simulations has Windows 10 with 8 GB Ram and intel(R) core(TM) i7-8700 CPU@3.20GHZ processor. All the experiments were performed in MATLAB 2013a (32-Bit) environment.

### 3.2. Parameters' Setting

- Population size npop = 100
- Number of generations Max Function Evalutions = 500000
- Number of runs = 25

### 3.3. Illustration of terms used for comparison

Illustration of the used terms in table for comparison is given in below.

- **Best:** It is the best fitness value for the obtained final best solutions in 25 runs.
- Median: It is the median fitness value for the obtained final best solutions in 25 runs.
- Worst: It is the Worst fitness value for the obtained final best solutions in 25 runs.
- c: It is the order triple representing the number of violated constraints in  $(1,\infty]$ , (0.01,1], and (0.0001, 0.01] respectively for median solution.
- $\overline{v}$ : It is the mean violation of median solution.
- Mean: It is the Mean fitness value for the obtained final best solution in 25 runs.
- St. Dev: It is the standard deviation of fitness value for the obtained final best solution in 25 runs.
- FR: It is the percentage of feasibility rate in 25 runs.
- SR: It is the percentage of success rate in 25 runs.

### 3.4. Comparison of Proposed Algorithms

Simulation results are compared for three proposed algorithms; NCVSF-TLBO, WMSF-TLBO, and HSF-TLBO, for CEC-2006 constrained suit consisting of 24 benchmark constrained problems. Details of the compared statical data are displayed in the tables below.

Prob No	Optimum	Algorithm	Best	Median	Worst	с	$\overline{v}$	Mean	St. Dev.	$\operatorname{FR}(\%)$	SR(%)
		NCVSF-TLBO	-15(0)	-10(0)	-6(0)	(0, 0, 0)	0	-10.2581	2.91799	100	12
G01	-15	WMSF-TLBO	-15(0)	-15(0)	-6(0)	(0, 0, 0)	0	-13.3781	2.27966	100	52
		HSF-TLBO	-15(0)	-15(0)	-9(0)	(0, 0, 0)	0	-13.4362	1.95619	100	52
		NCVSF-TLBO	-0.76395(0)	-0.592535(0)	-0.446209(0)	(0, 0, 0)	0	-0.589084	0.0823299	100	0
G02	-0.803619	WMSF-TLBO	-0.792607(0)	-0.592917(0)	-0.414959(0)	(0, 0, 0)	0	-0.589659	0.0851794	100	0
		HSF-TLBO	-0.803615(0)	-0.600225(0)	-0.45717(0)	(0, 0, 0)	0	-0.600881	0.0829211	100	4
		NCVSF-TLBO	-100000(1)	-100000(1)	-0.00154395(0)	(1, 0, 0)	9	-80000	40824.8	20	0
G03	-1.0005	WMSF-TLBO	-0.550806(0)	-0.0841718(0)	-0.00121948(0)	(0, 0, 0)	0	-0.146833	0.145905	100	0
		HSF-TLBO	-0.468707(0)	-0.179232(0)	-0.0143643(0)	(0, 0, 0)	0	-0.169795	0.103452	100	0
		NCVSF-TLBO	-30665.5(0)	-30665.5(0)	-30665.5(0)	(0, 0, 0)	0	-30665.5	3.85866e - 12	100	100
G04	-30665.5	WMSF-TLBO	-30665.5(0)	-30665.5(0)	-30665.5(0)	(0, 0, 0)	0	-30665.5	4.33006e - 12	100	100
		HSF-TLBO	-30665.5(0)	-30665.5(0)	-30665.5(0)	(0, 0, 0)	0	-30665.5	3.713e - 12	100	100
		NCVSF-TLBO	0(3)	0(3)	0(3)	(3, 0, 0)	542.665	0	0	0	0
G05	5126.5	WMSF-TLBO	5126.5(0)	5243.25(0)	5458.18(0)	(0, 0, 0)	0	5240.41	81.1682	100	4
		HSF-TLBO	5126.5(0)	5189.87(0)	5454.12(0)	(0, 0, 0)	0	5227.97	111.962	100	24
		NCVSF-TLBO	-7973(1)	-7973(1)	-6961.81(0)	(1, 0, 0)	5.5	-7730.32	440.766	24	24
G06	-6961.81	WMSF-TLBO	-6961.81(0)	-6961.81(0)	-6961.81(0)	(0, 0, 0)	0	-6961.81	3.713e - 12	100	100
		HSF-TLBO	-6961.81(0)	-6961.81(0)	-6961.81(0)	(0, 0, 0)	0	-6961.81	3.713e - 12	100	100
		NCVSF-TLBO	24.3064(0)	24.3102(0)	24.355(0)	(0, 0, 0)	0	24.3146	0.0113708	100	0
G07	24.3062	WMSF-TLBO	24.3065(0)	24.3108(0)	24.3396(0)	(0, 0, 0)	0	24.3129	0.00760074	100	0
		HSF-TLBO	24.3063(0)	24.3081(0)	24.3894(0)	(0, 0, 0)	0	24.3138	0.0169507	100	4
		NCVSF-TLBO	-0.095825(0)	-0.095825(0)	-0.095825(0)	(0, 0, 0)	0	-0.095825	0	100	100
G08	-0.095825	WMSF-TLBO	-0.095825(0)	-0.095825(0)	-0.095825(0)	(0, 0, 0)	0	-0.095825	2.83279e - 18	100	100
		HSF-TLBO	-0.095825(0)	-0.095825(0)	-0.095825(0)	(0, 0, 0)	0	-0.095825	2.83279e-18	100	100
		NCVSF-TLBO	680.63(0)	680.63(0)	680.637(0)	(0, 0, 0)	0	680.63	0.00151257	100	84
G09	680.63	WMSF-TLBO	680.63(0)	680.63(0)	680.63(0)	(0, 0, 0)	0	680.63	5.33567e - 07	100	100
		HSF-TLBO	680.63(0)	680.63(0)	680.63(0)	(0, 0, 0)	0	680.63	2.73025e - 07	100	100
		NCVSF-TLBO	2100(1)	2100(1)	2100(1)	(1, 0, 0)	204167	2100	0	0	0
G10	7049.25	WMSF-TLBO	7049.25(0)	7049.32(0)	7250.97(0)	(0, 0, 0)	0	7073.52	66.8781	100	0
		HSF-TLBO	7049.25(0)	7049.32(0)	7250.93(0)	(0, 0, 0)	0	7057.94	40.2647	100	4

Table 2: Comparison of Proposed Algorithms for CEC 2006 [26]

Prob No	Optimum	Algorithm	Best	Median	Worst	с	$\overline{v}$	Mean	St. Dev.	FR(%)	SR(%)
G11	0.7499	NCVSF-TLBO	0.749923(0)	0.755674(0)	0.928021(0)	(0, 0, 0)	0	0.773693	0.044111	100	4
		WMSF-TLBO	0.7499(0)	0.7499(0)	0.7499(0)	(0, 0, 0)	0	0.7499	1.13312e - 16	100	100
		HSF-TLBO	0.7499(0)	0.7499(0)	0.7499(0)	(0, 0, 0)	0	0.7499	1.13312e - 16	100	100
		NCVSF-TLBO	-1(0)	-1(0)	-1(0)	(0, 0, 0)	-0	-1	0	100	100
G12	-1	WMSF-TLBO	-1(0)	-1(0)	-1(0)	(0, 0, 0)	-0	$^{-1}$	0	100	100
		HSF-TLBO	-1(0)	-1(0)	-1(0)	(0, 0, 0)	-0	$^{-1}$	0	100	100
		NCVSF-TLBO	5.22653e - 76(3)	9.81492e - 65(2)	0.940811(2)	(2, 0, 0)	23.1865	0.0525531	0.197839	0	0
G13	0.0539415	WMSF-TLBO	0.158596(0)	0.535777(0)	0.992503(0)	(0, 0, 0)	0	0.561262	0.226208	100	0
		HSF-TLBO	0.0551461(0)	0.475098(0)	0.959623(0)	(0, 0, 0)	0	0.502159	0.258825	100	0
	-47.7649	NCVSF-TLBO	-2096.03(3)	-2096.03(3)	-2096.03(3)	(3, 0, 0)	58.6667	-2096.03	1.39237e - 12	0	0
G14		WMSF-TLBO	-47.4581(0)	-45.8565(0)	-41.7143(0)	(0, 0, 0)	0	-45.4412	1.51837	100	0
		HSF-TLBO	-47.7085(0)	-45.3164(0)	-42.1655(0)	(0, 0, 0)	0	-45.3344	1.45773	100	0
		NCVSF-TLBO	400(2)	400(2)	974.911(1)	(2, 0, 0)	254.5	445.465	157.371	0	0
G15	961.715	WMSF-TLBO	961.715(0)	961.715(0)	963.59(0)	(0, 0, 0)	0	961.824	0.389466	100	76
		HSF-TLBO	961.715(0)	961.715(0)	963.447(0)	(0, 0, 0)	0	961.83	0.397135	100	88
	-1.90516	NCVSF-TLBO	-2.0965(1)	-1.90516(0)	-1.90514(0)	(0, 0, 0)	0	-1.98169	0.0956748	60	60
G16		WMSF-TLBO	-1.90516(0)	-1.90516(0)	-1.90516(0)	(0, 0, 0)	0	-1.90516	9.81569e-16	100	100
		HSF-TLBO	-1.90516(0)	-1.90516(0)	-1.90516(0)	(0, 0, 0)	0	-1.90516	8.01875e-16	100	100
	8853.54	NCVSF-TLBO	8213.33(4)	8213.33(4)	8717.17(3)	(4, 0, 0)	668.685	8233.48	100.768	0	0
G17		WMSF-TLBO	8857.39(0)	8942.7(0)	9234.29(0)	(0, 0, 0)	0	8988.52	128.138	100	0
		HSF-TLBO	8857.33(0)	8939.84(0)	9210.4(0)	(0, 0, 0)	0	8960.6	104.235	100	0
G18	-0.866025	NCVSF-TLBO	-13.6421(4)	-0.866024(0)	-0.674973(0)	(0, 0, 0)	0	-1.55513	2.63514	84	56
		WMSF-TLBO	-0.866025(0)	-0.866009(0)	-0.860728(0)	(0, 0, 0)	0	-0.865376	0.00138706	100	60
		HSF-TLBO	-0.866025(0)	-0.866024(0)	-0.863875(0)	(0, 0, 0)	0	-0.865751	0.000645271	100	80
	32.6556	NCVSF-TLBO	32.6558(0)	32.6613(0)	46.1546(0)	(0, 0, 0)	0	33.8854	2.94461	100	0
G19		WMSF-TLBO	32.6558(0)	32.6613(0)	46.1559(0)	(0, 0, 0)	0	33.443	2.75087	100	0
		HSF-TLBO	32.6557(0)	32.6597(0)	35.3435(0)	(0, 0, 0)	0	32.8748	0.740456	100	4
		NCVSF-TLBO	1.02608e - 97(3)	2.08294e - 97(3)	0.0694016(3)	(1, 2, 0)	0.145683	0.00277606	0.0138803	0	0
G20	0.204979	WMSF-TLBO	0.0801573(1)	0.259324(1)	0.981122(3)	(0, 1, 0)	0.0243059	0.339315	0.197291	0	0
		HSF-TLBO	0.0469487(2)	0.180092(1)	1.10409(2)	(0, 1, 0)	0.02787	0.269931	0.250122	0	0
G21	193.725	NCVSF-TLBO	0(3)	0(4)	309.66(3)	(2, 2, 0)	333.698	12.3864	61.932	0	0
		WMSF-TLBO	193.725(0)	324.986(0)	906.837(0)	(0, 0, 0)	0	386.296	193.77	100	16
		HSF-TLBO	193.725(0)	193.755(0)	999.999(0)	(0, 0, 0)	0	309.366	234.237	96	48
G22	236.431	NCVSF-TLBO	0(12)	0(12)	3998.1(10)	(9, 3, 0)	2.65004e + 06	159.924	799.62	0	0
		WMSF-TLBO	141.833(13)	9832.31(19)	20000(3)	(14, 5, 0)	2.06585e + 06	10754.5	5523.94	0	0
		HSF-TLBO	0(9)	19917.4(7)	20000(6)	(5, 2, 0)	442453	13928.5	8229.91	0	0
G23	-400.055	NCVSF-TLBO	-0.0009(0)	0(0)	0(0)	(0, 0, 0)	0	-3.6e-05	0.00018	100	0
		WMSF-TLBO	-399.937(0)	-100.046(0)	-0.295254(0)	(0, 0, 0)	0	-128.008	145.137	100	0
		HSF-TLBO	-400.055(0)	-77.9528(0)	-0.0766(0)	(0, 0, 0)	0	-134.04	154.896	100	16
		NCVSF-TLBO	-5.50801(0)	-5.50801(0)	-5.50801(0)	(0, 0, 0)	0	-5.50801	2.53818e - 15	100	100
G24	-5.50801	WMSF-TLBO	-5.50801(0)	-5.50801(0)	-5.50801(0)	(0, 0, 0)	0	-5.50801	2.45925e - 15	100	100
		HSF-TLBO	-5.50801(0)	-5.50801(0)	-5.50801(0)	(0, 0, 0)	0	-5.50801	2.60843e - 15	100	100

Table 3: Comparison of Proposed Algorithms for CEC 2006 [26]

# 3.5. Fitness and feasibility convergence graphs

Fitness and feasibility convergence graphs for representing problems are displayed for Proposed Algorithms in following:



Figure 2: Fitness and feasibility convergence graphs of G01



Figure 3: Fitness and feasibility convergence graphs of  $\mathrm{G02}$ 



Figure 4: Fitness and feasibility convergence graphs of  $\mathrm{G03}$ 



Figure 5: Fitness and feasibility convergence graphs of  $\mathrm{G04}$ 



Figure 6: Fitness and feasibility convergence graphs of G05



Figure 7: Fitness and feasibility convergence graphs of  $\mathrm{G06}$ 



Figure 8: Fitness and feasibility convergence graphs of G07



Figure 9: Fitness and feasibility convergence graphs of  $\mathrm{G08}$ 



Figure 10: Fitness and feasibility convergence graphs of  $\mathrm{G09}$ 



Figure 11: Fitness and feasibility convergence graphs of  $\operatorname{G10}$ 



Figure 12: Fitness and feasibility convergence graphs of G11  $\,$ 



Figure 13: Fitness and feasibility convergence graphs of  $\mathrm{G12}$ 



Figure 14: Fitness and feasibility convergence graphs of G13  $\,$ 



Figure 15: Fitness and feasibility convergence graphs of  $\mathrm{G14}$ 



Figure 16: Fitness and feasibility convergence graphs of  $\mathrm{G15}$ 



Figure 17: Fitness and feasibility convergence graphs of  ${\rm G16}$ 



Figure 18: Fitness and feasibility convergence graphs of G17



Figure 19: Fitness and feasibility convergence graphs of  $\operatorname{G18}$ 



Figure 20: Fitness and feasibility convergence graphs of G19  $\,$ 



Figure 21: Fitness and feasibility convergence graphs of G20



Figure 22: Fitness and feasibility convergence graphs of G21



Figure 23: Fitness and feasibility convergence graphs of  $\mathrm{G22}$ 



Figure 24: Fitness and feasibility convergence graphs of  ${\rm G23}$ 



Figure 25: Fitness and feasibility convergence graphs of G24

### 3.6. Discussion

In this paper, three CHTs named NCVSF, WMSF, and HSF based on the concept of SF are implemented and integrated with TLBO. The three constrained variants of TLBO, namely, NCVSF-TLBO, WMSF-TLBO, and HSF-TLBO are applied for checking robustness, consistency, and efficiency to solve the CEC 2006 benchmark functions[26]. The Statistical analysis, enable us to state the following discussion in respect of developed algorithms.

- **G01**: For this problem HSF-TLBO is leading algorithm since it is comparable or defeating the remaining based on all statistics used for evaluation.
- **G02**: For this problem HSF-TLBO is leading algorithm since statistics displayed that except St.Dev. in all remaining it is winner or comparable.
- **G03:** For this problem HSF-TLBO is leading algorithm since statistics analysis displayed that except Best in all remaining it is winner or comparable.
- **G04:** For this problem proposed algorithms are comparable in all statistics except St.Dev. On the basic of St.Dev. we can say that HSF-TLBO is winner.
- **G05**: For this problem HSF-TLBO is leading algorithm since statistics displayed that except St.Dev. in all remaining it is winner or comparable.
- **G06:** For this problem HSF-TLBO and WMSF-TLBO are leading and comparable algorithms since statistics are same for these two.
- G07: For this problem for statistics Best, Median, and SR HSF-TLBO is winner for c and  $\overline{v}$ , and FR all comparable. For remaining statistics the performance of HSF-TLBO is not notable.
- **G08:** For this problem proposed algorithms are comparable in all statistics except St.Dev. On the basic of St.Dev. we can say that NCVSF-TLBO is winner since other statistics are matched for all.
- **G09:** For this problem HSF-TLBO is leading algorithm since it is comparable or defeating the remaining based on all statistics used for evaluation.
- **G10:** For this problem HSF-TLBO is leading algorithm since it is comparable or defeating the remaining based on all statistics used for evaluation.
- **G11:** For this problem HSF-TLBO and WMSF-TLBO are leading and comparable algorithms since statistics are same for these two.
- **G12:** For this problem proposed algorithms are comparable since statistics are same for these.
- **G13:** For this problem HSF-TLBO is leading algorithm since it is comparable or defeating the remaining based on all statistics used for evaluation.
- G14: For this problem HSF-TLBO is comparable or winner in all statistics except median and mean. It is very closed to the winner in these statistics.
- G15: For this problem HSF-TLBO is comparable or winner in all statistics except mean and St.Dev. It is very closed to the winner in these statistics.
- **G16:** For this problem after a very closed competition, HSF-TLBO leaded over WMSF-TLBO based on St.Dev.

- **G17:** For this problem HSF-TLBO exclusively leaded over the remaining algorithms based on almost all statistics.
- **G18:** For this problem HSF-TLBO exclusively defeated the remaining algorithms based on almost all statistics with high SR value.
- **G19**: For this problem HSF-TLBO exclusively defeated the remaining algorithms based on almost all statistics with some SR value; for while remaining the SR values is zero.
- **G20**: This is a hard problem of the suit; therefore, performance of the proposed algorithms are not notable.
- G21: For this problem HSF-TLBO and WMSF-TLBO are very closed in competition, but one can declared HSF-TLBO based on its SR values which is very high as compared to WMSF-TLBO.
- G22: This is the hard problem of the suit; therefore, performance of the proposed algorithms are not notable, but based on low infeasibility the claim of declaring HSF-TLBO better as compared to the remaining is legal.
- **G23:** This is also a hard problem of the suit, but based on high value of SR HSF-TLBO can be declared as winner.
- **G24**: For this problem proposed algorithms are comparable in all statistics except St.Dev. based on it WMSF-TLBO can be declared as winner.

### 4. Conclusion

Current research work brings in the concept of three constrained versions of TLBO, NCVSF-TLBO, WMSF-TLBO, and HSF-TLBO. In these algorithms, for handling the constraints of the problem the platform of superiority of feasibility is used. Where factors upon which the decision of winner is made are NCV, WM and hybrid of these i.e HSF. These are integrated in the framework of TLBO to solve COPs. For evaluation purposes CEC-2006 constrained benchmarks are used. Obtained results from simulations are compared, based on it the following notables points can be concluded:

- Success Rate (SR): Based on the SR values HSR-TLBO has winning status since no algorithms among compared existed that can beat it at any problem of the suit.
- Feasibility Rate (FR): Based on the FR values HSF-TLBO and WMSF-TLBO are comparable for almost all problem except G21.
- **Best Fitness:** Based on the Best fitness values HSF-TLBO is leading algorithm on average problems i.e winner for the competition based on Best fitness value.

- Median Fitness: Based on the Median fitness values HSF-TLBO is leading algorithm on almost all problems i.e winner for the completion based on Median fitness value.
- Worst Fitness: Based on the Worst fitness values HSF-TLBO is leading algorithm on almost all problems i.e winner for the completion based on Worst fitness value.
- c: Based on the the order triple value c HSF-TLBO and WMSF-TLBO are comparable for whole suit.
- $\overline{v}$ : Based on the the  $\overline{v}$  value HSF-TLBO and WMSF-TLBO are comparable for most problem, but; there exist some problem based on them on can say HSF-TLBO is in prominent status as compared to WM-TLBO.
- Mean Fitness: Based on the Mean fitness values HSF-TLBO is leading algorithm on most problems i.e winner for the completion based on Mean fitness value.
- **St.Dev:** Based on the St.Dev values HSF-TLBO is winning the competition since it beat the others compared.
- Over All Conclusions: Over all conclusion about the proposed algorithms is that in most problems HSR-TLBO in winner; therefore, it is the consistent, robust, and prominent among the compared.

We will work in future keeping in veiw the following tasks:

- To check efficiency and robustness of HSF in other existed NIAs environments.
- To extend the researched idea for solving other constrained problems.
- To do parameters adjustment for specific type of problems to insure that certain setting is suitable for certain type problems.
- To check the algorithms efficiency when the order of the implantation of teacher phase and learner phase are reversed.

## References

- A. Antoniou and W.-S. Lu, Practical Optimization: Algorithms and Engineering Applications. Springer Science & Business Media, 2007.
- [2] G. Venter, "Review of Optimization Techniques," 2010.
- [3] A. Törn and A. Žilinskas, *Global optimization*. Springer, 1989, vol. 350.
- [4] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, 1st ed. John Wiley and Sons, Chichester, UK, 2001.

- [5] J. R. Koza and R. Poli, "Genetic programming," in Search Methodologies. Springer, 2005, pp. 127–164.
- [6] X. Yu and M. Gen, Introduction to evolutionary algorithms. Springer Science & Business Media, 2010.
- [7] —, Introduction to Evolutionary Algorithms, R. Roy, Ed. Springer London Dordrecht Heidelberg New Yark, 2011.
- [8] A. ur Rehman, A. Islam, and S. B. Belhaouari, "Multi-Cluster Jumping Particle Swarm Optimization for Fast Convergence," *IEEE Access*, vol. 8, pp. 189382–189394, 2020.
- H. Garg, "A hybrid gsa-ga algorithm for constrained optimization problems," *Information Sciences*, vol. 478, pp. 499 – 523, 2019.
- [10] M. Mareli and B. Twala, "An adaptive cuckoo search algorithm for optimisation," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 107–115, 2018.
- [11] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, second edition ed. Springer Publishing Company, Incorporated, 2015.
- [12] Y. Y. AO, "An Adaptive Differential Evolution Algorithm to Solve Constrained Optimization Problems in Engineering Design," *Engineering*, vol. 02, no. 01, pp. 65–77, 2010.
- [13] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: Methodologies, architectures, and reviews," 2007.
- [14] F. H. M. T. Back and H. Schwefel, "A Survey of Evolution Strategies," Proceeding of the Fourth International Conference on "Genetic Algorithms", Mateo, CA: Morgan Kauffman.
- [15] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. New York, NY, USA: Oxford University Press, Inc., 1996.
- [16] H.-G. Beyer and H.-P. Schwefel, "Evolution Strategies-A Comprehensive Introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [17] Y. M. Xie and G. P. Steven, "A simple evolutionary procedure for structural optimization," Computers & structures, vol. 49, no. 5, pp. 885–896, 1993.
- [18] J. H. Holland, "Genetic Algorithms and the Optimal Allocation of Trials," SIAM J. Comput., vol. 2, no. 2, pp. 88–105, 1973.
- [19] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1<sup>st</sup> ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

- [20] J. R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, 1992.
- [21] J. Koza, "Genetic programming II Autonomous Discovery of Reusable Programe," MIT Press, 1995.
- [22] R. Poli, W. B. Langdon, and N. F. McPhee, A Field Guide to Genetic Programming. Lulu Enterprises, UK Ltd, 2008.
- [23] R. V. Rao, "Teaching-learning-based optimization algorithm," in *Teaching learning based optimization algorithm*, 2016, pp. 9–39.
- [24] N. Marjanovic, N. Kostic, N. Petrovic, M. Blagojevic, and M. Matejic, "Teaching-learning-based optimization algorithm for solving machine design constrained optimization," Annals of the Faculty of Engineering Hunedoara, vol. 15, no. 2, p. 105, 2017.
- [25] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 561–579, 2010.
- [26] J. Liang, T. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization," 2006.