\( \varepsilon \)-constraint heat transfer search (\( \varepsilon \)-HTS) algorithm for solving multi-objective engineering design problems

Mohamed A. Tawhid\(^1,2\) and Vimal Savsani\(^3,4\)

\(^1\)Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC, Canada V2C 0C8 (Email: mtawhid@tru.ca)

\(^2\)Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Moharam Bey 21511, Alexandria, Egypt

\(^3\)Department of Mechanical Engineering, Pandit Deendayal Petroleum University, Gandinagar, Gujarat, India

\(^4\)Postdoctoral Fellow, Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC, Canada V2C 0C8 (Email: vsavsani@tru.ca, vimal.savsani@gmail.com)

Abstract:

In this paper, an effective \( \varepsilon \)-constraint heat transfer search (\( \varepsilon \)-HTS) algorithm for the multi-objective engineering design problems is presented. This algorithm is developed to solve multi-objective optimization problems by evaluating a set of single objective sub-problems. The effectiveness of the proposed algorithm is checked by implementing it on multi-objective benchmark problems that have various characteristics of Pareto front such as discrete, convex, and non-convex. This algorithm is also tested for several distinctive multi-objective engineering design problems, such as four bar truss problem, gear train problem, multi-plate disc brake design, speed reducer problem, welded beam design, and spring design problem. Moreover, the numerical experimentation shows that the proposed algorithm generates the solution to represent true Pareto front.

Keywords: Multi-objective optimization, heat transfer search, design optimization, Pareto front
1. Introduction:

The objective of this paper is to solve optimization problems which have more than one objective functions. These optimization problems are recognized as multi-objective optimization problems (MOOPs). Solving MOOPs is very strenuous and difficult task because their multiple objective functions are often conflicting. MOOPs are significant tools in solving many different engineering disciplines, finance, management science, economics, and many other disciplines (Goicoechea et al. 1982; Andersson, 2000; Coello et al., 2005; Marler & Arora, 2004; Li and Zhang, 2009). Several applications of MOOPs exist in engineering that help the decision maker or the design engineer to select an appropriate system or design from the available set of solutions obtained as a Pareto front. It is not possible to list all the applications of MOOPs in this work, but many such latest applications (Ahmadi et al., 2015; Sadatsakkak e al., 2015; Ahmadi et al., 2016a; Ahmadi et al., 2016b; Qu et al., 2016; Ahmadi et al., 2016c; Cavaliere et al., 2016; Gadhavi et al., 2016; Bandaru et al., 2017; Zhou et al., 2011; Luna et al., 2010; Mondal et al., 2013; Moslehi & Mahnam, 2011) are available in the literature to highlight the importance of MOOPs in engineering regime.

Although, single objective optimization problems (SOOPs) have a single objective to be optimized at a given time and the solution of which is a single optimal point normally, the optimal solution in MOOPS that concurrently optimizes all the objective functions does not exist.

Thus, the decision makers for solving MOOPs are acquisitive in taking the “most preferred” solution in contrast to the optimal solution and find the set of best trade-offs between the objective functions. The Pareto optimal (or non-inferior, non-dominated, efficient) solutions are the solutions that cannot be enhanced in one objective function without worsening their performance in at least one of the rest. The set of the Pareto optimal solutions is the Pareto set. Most of all the methods for solving MOOPs have aimed to find the Pareto front solutions as near as possible to the optimal Pareto front and maintain variation among the optimal set of solutions.

Weighted sum method is the simplest popular method that solves multi-objective optimization problem by solving different single objective sub-problems (Ehrgott and Gandibleux, 2002). These sub-problems are generated by the linear combination of the objectives. Various combinations of the weights generate various non-dominated solutions. The main drawback of this method is that it cannot find the solution for the non-convex objective space. Also, it is very difficult to solve a problem in which the value of both the objectives is significantly different.

Several researchers have overcome the disadvantages of the weighted sum method and solve problems with non-convex objective space by ε-constraint method (Khalili and Amiri, 2012; Aghaei et al., 2011). This method also formulates the multi-objective problem in to different single objective sub-problems by converting all the objectives in the form of constraints except one. The objectives which are covered in to constraints are restricted by the different values of ε,
and so this method is known as $\varepsilon$-constraint method. Different values of $\varepsilon$ produces different non-dominated solutions in the objective space.

Recently, many researchers have developed and proposed successful and powerful algorithms in finding the global optimum of complex problems of MOOPs, called evolutionary algorithms (EAs). Such algorithms to solve MOOPs are known as multi-objective evolutionary algorithms (MOEAs) when they use EAs as their basic search techniques. Many such methods which works on the strategy of EAs are; vector evaluated genetic algorithms (VEGA) (Schaffer, 1985; Zitzler and Thiele, 1999), multi-objective genetic algorithm (MOGA) (Knowles and Corne, 1999), non-dominated sorting genetic algorithm (NSGA) (Srinivas and Deb, 1994), niched Pareto genetic algorithm (NPGA) (Corne et al., 2001), strength Pareto evolutionary algorithm (SPEA2 and SPEA) (Zhou et al., 2011; Zitzler and Thiele, 1999; Zitzler et al., 2001), Pareto archived evolution strategy (PAES) (Knowles and Corne, 1999), Pareto envelope-based selection algorithm (PESA and PESA-II) (Corne et al., 2001), an elitist based non-dominated sorting genetic algorithm – II (NSGA-II) (Deb et al., 2002), a multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang and Li, 2007) and its improved versions described in detail in (Zhou et al., 2011). Apart from EAs, swarm intelligence based multi-objective optimization were developed and have been employed and addressed to a variety of MOOPs (Coello and Lechuga, 2002; Mostaghim and Teich, 2004; Agrawal et al., 2008). Several other multi-objective algorithms were developed that utilizes the search techniques of particle swarm optimization (PSO) (Reyes and Coello, 2006; Moslehi and Manham, 2011; Wang and Yang, 2009), artificial bee colony (ABC) (Omkar et al., 2011; Akbari et al., 2012; Zhang et al., 2012), artificial immune system algorithm (AISA) (Tan et al., 2008), ant colony optimization (ACO) (Angus, 2009; Yagmahan and Yenisey, 2008), artificial immune algorithm (AIA) (Aydin, 2011; Gong et al., 2008), gravitational search algorithm (GSA) (Mondal et al., 2013), biogeography-based optimization (BBO) (Jamuna and Swarup, 2012; Roy et al., 2010), invasive weed optimization (IWA) (Nikoofard et al., 2012), firefly algorithm (FFA) (Yang, 2013), cuckoo search algorithm (CSA) (Yang and Deb, 2013), bat algorithm (BA) (Yang, 2011) and teaching-learning based optimization (TLBO) (Krishnanand et al., 2011; Patel and Savsani; 2014a, 2014b). All such methods have shown successful evidence in getting the Pareto front closest to the true Pareto front.

HTS is an effective meta-heuristic developed in 2015 (Patel and Savsani, 2015). This method has proved its effectiveness and robustness in terms of accuracy, computational efforts and convergence for the single objective optimization. HTS works on the natural phenomena of heat transfer by conduction, convection and radiation to maintain the equilibrium with the surrounding. This paper is mainly focused on to develop a novel technique by using $\varepsilon$-constraint multi-objective optimization method along with a recently developed meta-heuristics called heat transfer search (HTS). This proposed method will be denoted as $\varepsilon$ – HTS method. Later in this paper it is shown that the $\varepsilon$ – HTS can generate non-dominated solutions and true Pareto front (PF). The first contribution of this paper is to introduce a new concept of $\varepsilon$-constraint
method with the HTS algorithm. Second contribution is to show the correctness of $\epsilon$-HTS method for the engineering design problems. The proposed method is investigated on different multi-objective benchmark functions and engineering design problems. The proposed approach is compared with the other well known multi-objective optimization algorithms.

The organization of this paper is as follows: Section 2 defines multi-objective optimization and Pareto concepts. Section 3 describes the heat transfer search (HTS) algorithm in details. Section 4 is focused to cover the basic of $\epsilon$-HTS algorithm. Section 5 investigates the performance of $\epsilon$-HTS on different benchmark problems followed by section 6 which investigates the performance of $\epsilon$-HTS or different engineering design problems. Section 7 gives the general conclusion.

2. Multi-objective optimization problems:

A general multi-objective problem consists of several objective functions and is associated to several constraints. Mathematically, the problem can be formulated as (Berube et al., 2009, Laumanns et al., 2006; Miettinen, 2012):

$$\text{min } f(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}) \ldots f_n(\bar{x})], \; \bar{x} = \{x_1, x_2, \ldots, x_k\} \in S$$

Subjected to: $g_i(\bar{x}) \leq 0, i = 1, 2, \ldots I$

$h_j(\bar{x}) = 0, j = 1, 2, \ldots J$

$(\bar{x})^l \leq \bar{x} \leq (\bar{x})^u$

where, $S$ is the set of feasible solutions referred as solution space. $f_1(\bar{x}), f_2(\bar{x}) \ldots f_n(\bar{x})$ are individual objective functions, $g_i(\bar{x})$ are inequality constraints and $h_j(\bar{x})$ are equality constraints, $(\bar{x})^l$ and $(\bar{x})^u$ are bound constraints indicating lower and upper limits for the design variables.

The objective space is defined by $\mathcal{F} = \{\bar{f} = (f_1, f_2, \ldots f_n): f_i = f_i(\bar{x}), \forall \bar{x} \in S, i = 1, 2, \ldots n\}$. Since there is no solution that optimizes at the same times all objectives, one will search for a suitable trade-off instead of an optimal solution. Although some solutions might be considered as equivalent, this trade-off must be such that no strictly better solution exists, though. This involves a partial order of the objective space, defined by a ‘dominance relation’ which is used to characterize ‘Pareto optimality or efficiency’. Let $\bar{f}^1$ and $\bar{f}^2 \in \mathcal{F}$, $\bar{f}^1$ dominates $\bar{f}^2$ (denoted as $\bar{f}^1 > \bar{f}^2$), if and only if $f^1_i \leq f^2_i, i = 1, 2, \ldots n$, where at least one inequality is strict.

$\bar{x} \in S$ is Pareto optimal (non-dominated) in $S$, if and only if $\nexists \bar{x}' \in S$ such that $f(\bar{x}') > f(\bar{x})$. The set of all Pareto-optimal design variables is called Pareto set. The Pareto set $PS = \{\bar{x} \in S: \bar{x} \text { is Pareto optimal in } S\}$. 

Page 4 of 46
\( PF = \{ f(\vec{x}) : \vec{x} \in PS \} \) is denoted to Pareto front.

It can be noted from that the PS is defined on the solution space whereas PF is defined on the objective space.

3. **Heat transfer search (HTS) algorithm:**

Patel and Savsani (2015) proposed the HTS algorithm which is based on the natural law of thermodynamics and the principle of thermal equilibrium of the system. Thus, thermodynamically imbalanced systems always try to attain thermal equilibrium by initiating heat transfer between the system and its surrounding. The modes of heat transfer (Viz. conduction, convection, and radiation) play an essential role to set the thermal equilibrium. Therefore, the HTS algorithm considers ‘the conduction phase’, ‘the convection phase’, and ‘the radiation phase’ to reach an equilibrium state. In the HTS algorithm, all three modes of heat transfer have equal probability to transfer heat and one of the heat transfer mode is decided randomly for each generation during the course of optimization.

The HTS algorithm is a population-based algorithm that initiates with a randomly generated population, where the system has ‘\( n \)’ number of molecules (i.e. population size) and the temperature level (i.e. design variables) is ‘\( m \)’. In the next stage, the population is updated in each generation ‘\( g \)’ by one of the randomly selected heat transfer mode. Furthermore, the updated solution in the HTS algorithm is accepted only if it has a better functional value. Afterwards, the worst solutions of the population are replaced by the elite solutions and finally identical solutions are replaced by a randomly generated solution if it exists. Thus, the better solution can be obtained by performing the difference between the current solution and either of the best solutions, another random solution, or the mean value of solutions from the population. The detailed of all three phases of the HTS algorithm is described as follows.

In the conduction phase, heat transfer happens because of the conduction between molecules of the substance. Thus, more energetic molecules transfer heat to less energetic molecules to get a state of the thermal equilibrium. Conduction can also take place between the system and the surrounding when both are in direct physical contact with each other. Conduction heat transfer is governed by the Fourier’s law of heat conduction. The mathematical formulation of the updated molecule is given in Equation 1 and 2.

\[
T^\prime_{j,i} = \begin{cases} 
T_{k,i} + (-R^2 * T_{k,i}), & \text{if } F(T_j) > F(T_k) \\
T_{j,i} + (-R^2 * T_{j,i}), & \text{if } F(T_j) < F(T_k) \end{cases} ; \text{ if } g \leq g_{max}/CDF
\]

\[
T^\prime_{j,i} = \begin{cases} 
T_{k,i} + (-r_i * T_{k,i}), & \text{if } F(T_j) > F(T_k) \\
T_{j,i} + (-r_i * T_{j,i}), & \text{if } F(T_j) < F(T_k) \end{cases} ; \text{ if } g \geq g_{max}/CDF
\]

Where, \( T^\prime_{j,i} \) is the updated molecule; \( j = 1, 2, ..., n \); \( k \) is a randomly selected molecule; \( j \neq k \); \( k \in (1, 2, ..., n) \); \( i \) is a randomly selected design variable; \( i \in (1, 2, ..., m) \); \( FE \) is function.
evaluation; \(CDF\) is the conduction factor; \(R\) is the probability variable; \(R \in [0, 0.3333]\); \(r_i\) is a random number; \(r_i \in [0, 1]\); \(R^2\) and \(r_i\) represents the conductance parameters of the Fourier’s equation; \(T_j\) and \(T_k\) represents the temperature difference of molecules; and \(CDF\) is set to 2, it balances the exploration and exploitation capabilities of the conduction phase. In this phase, only one design variable is updated in each generation of the course of optimization.

In the convection phase, heat transfer happens because of convection between the system and the adjacent fluid in motion. Thus, the system temperature (the mean temperature) interacts with the adjacent fluid temperature (the surrounding) to reach a state of the thermal equilibrium. The best solution is considered as the surrounding. Convection heat transfer is stated by the Newton’s law of cooling. This phase is presented in Equations 3 and 4.

\[
T_{j,i}' = T_{j,i} + R \times (T_s - T_{ms} \times TCF)
\]

\[
TCF = \begin{cases} 
\text{abs}(R - r_i), & \text{if } g \leq g_{max}/COF \\
\text{round}(1 + r_i), & \text{if } g \geq g_{max}/COF 
\end{cases}
\]

Where, \(T_{j,i}'\) is the updated population; \(j = 1, 2, ... , n; i = 1, 2, ... , m; FE\) is function evaluation; \(COF\) is the convection factor; \(R\) is the probability variable; \(R \in [0.6666, 1]\); \(r\) is a random number; \(r_i \in [0, 1]\); \(R\) and \(r\) represent the convection parameters of Newton’s law of cooling; \(T_s\) and \(T_{ms}\) represents the temperature difference of surrounding temperature and the mean temperature of the system, respectively; \(TCF\) is a temperature change factor, it balances the exploration and exploitation capabilities of the convection phase; and \(COF\) is set to 10. In this phase, all the design variables are updated in each generation of the course of optimization.

In the radiation phase, heat transfer happens because of radiation emitted in the form of electromagnetic waves (or photons) because of its temperature level. Thus, the system interacts with the surrounding temperature (i.e. best solution) or within the system (i.e. other solution) to get a state of the thermal equilibrium. All bodies at a temperature above absolute zero emanate radiations. The maximum rate of radiation heat transfer determined by the absolute temperature level and described by the Stefan-Boltzmann law. The updated population is given in Equations 5 and 6.

\[
T_{j,i}' = \begin{cases} 
T_{j,i} + R \times (T_{k,i} - T_{j,i}), & \text{if } F(T_j) > F(T_k) \\
T_{j,i} + R \times (T_{j,i} - T_{k,i}), & \text{if } F(T_j) < F(T_k) 
\end{cases}
\]

\[
T_{j,i}' = \begin{cases} 
T_{j,i} + r_i \times (T_{k,i} - T_{j,i}), & \text{if } F(T_j) > F(T_k) \\
T_{j,i} + r_i \times (T_{j,i} - T_{k,i}), & \text{if } F(T_j) < F(T_k) 
\end{cases}
\]

Where, \(T_{j,i}'\) is the updated population; \(j = 1, 2, ... , n; i = 1, 2, ... , m; j \neq k; k \in (1, 2, ..., n); g\) is the current generation and \(g_{max}\) is the maximum number of generation specified; \(k\) is randomly selected molecules; \(R\) is the probability variable; \(R \in [0.3333, 0.6666]\); \(R\) and \(r\) represent the
radiation parameters of the Stefan-Boltzmann law; \( r_i \) is a random number; \( r_i \in [0, 1] \); 
\( T_\square \) and \( T_k \) represents temperature of a system and the surrounding, respectively; RDF is the radiation factor and RDF is set to 2, it balances the exploration and exploitation capabilities of the radiation phase. In this phase, all the design variables are updated in each generation of the course of optimization.

The HTS algorithm runs in three phases such as ‘the conduction phase’, ‘the convection phase’, and ‘the radiation phase’. Besides, each phase is divided into two sub-phases, which is controlled by function evaluations and various factors of heat transfer phases. Thus, the value of design variables can move small or large as all three modes of heat transfer have equal probability to transfer heat. The large and small change of the design variables represents the exploration and exploitation of a search space, respectively. The procedure for the basic HTS is given in Algorithm-1.

**Algorithm 1: basic Heat Transfer Search (HTS)**

```plaintext
Initialize population size (n), Number of design variables (m), limits on design variables (L, U), stopping criteria (FE_{max}, g_{max}) CDF, COF, RDF /\* initialization /\*

\[ T_{j,i} = L_{j,i} + \text{rand} \times (U_{j,i} - L_{j,i}), \text{for } j \in [1,n], \text{for } i \in [1,m] /\* initialize population /\*

Evaluate the population and arrange the population in ascending order \( \rightarrow FE = n \)

While ( \( g < g_{max} \) and \( FE < FE_{max} \) ) /\* begin the optimization loop /\*

\( R = \text{rand} \in [0, 1] \) /\* R decides the probability for the selection of phases /\*

\[ \text{for } j = 1 : n \]

\[ \text{if } R < \frac{1}{3} \]

\[ k \in [1, n] \neq j \]

\[ i \in [1, m] \]

\[ \text{if } F(T_j) > F(T_k) \]

\[ \text{if } g < g_{max} /\* \text{CDF } \text{to } T_{j,i}' = T_{j,i} + (-R^2 \times T_{k,i}) \]

\[ \text{else } \]

\[ T_{j,i}' = T_{j,i} + (-r_i \times T_{k,i}) \]

\[ \text{Evaluate } f(T_j') \rightarrow FE = FE + 1 \]

\[ f(T_j') < f(T_j) \leftrightarrow T_j = T_j' \] /\* greedy selection /\*

\[ \text{else if } g < g_{max} /\* \text{CDF } \text{to } T_{k,i}' = T_{j,i} + (-R^2 \times T_{j,i}) \]

\[ \text{else } \]

\[ T_{k,i}' = T_{j,i} + (-r_i \times T_{j,i}) \]

\[ \text{Evaluate } f(T_k') \rightarrow FE = FE + 1 \]

\[ f(T_k') < f(T_k) \leftrightarrow T_k = T_k' \] /\* greedy selection /\*

\[ \text{end if} \]

\[ \text{else if } 0.3333 < R \land R < 0.6666 \]

\[ k \in [1, n] \neq j \]

\[ \text{if } F(T_j) > F(T_k) \]

\[ \text{if } g < g_{max} /\* \text{CDF } \text{to } T_{j,i}' = T_{j,i} + R * (T_{k,i} - T_{j,i}) \]

\[ \text{else } \]

\[ T_{j,i}' = T_{j,i} + r_i * (T_{k,i} - T_{j,i}) \]

\[ \text{else } \]

\[ \text{if } g < g_{max} /\* \text{CDF } \text{to } T_{k,i}' = T_{j,i} + R * (T_{j,i} - T_{k,i}) \]

\[ \text{else } \]

\[ T_{j,i}' = T_{j,i} + r_i * (T_{j,i} - T_{k,i}) \]

\[ \text{endif} \]

\[ \text{Evaluate } f(T_j') \rightarrow FE = FE + 1 \]
```

Page 7 of 46
else
    \[ f(T'_j) < f(T_j) \iff T_j = T'_j \]  
    /* greedy selection*/
  end if

if \[ g \leq g_{\text{max}}/\text{COF} \] \to \forall i: \[ T'_{j,i} = T_{j,i} + R * (T_{j,i} - T_{j,\text{mean}} * \text{abs}(R - r_i)) \]
else \to \forall i: \[ T'_{j,i} = T_{j,i} + R * (T_{j,i} - T_{j,\text{mean}} * \text{round}(1 + r_i)) \]

Evaluate \[ f(T'_j) \to FE = FE + 1 \]

end for

for \( j = 1:2:n \)
  /*remove duplicate solution*/
  
  \[ \text{If} T_j = T_{j+1} \to T'_{j+1,i} = L_i + \text{rand}(U_i - L_i), \text{not} \forall i \to FE = FE + 1 \]
end for

end while

4. \( \epsilon \) – constraint Heat transfer search method (\( \epsilon \) – HTS) for multi-objective optimization:

The \( \epsilon \) – constraint method solves the multi-objective optimization problems by converting the multi-objective problem in the form of single objective sub-problems. It solves \( E_k(\epsilon_j) \) \( \epsilon \) – constraint problems. The problem \( E_k(\epsilon_j) \) can be stated as:

\[
\min f_i(\bar{x})
\]

Subjected to: \( f_j(\bar{x}) \leq \epsilon_j, i \neq j, i = 1,2, ..., n \)

**Theorem 1.** \( \bar{x}' \) is a Pareto optimal solution if and only if \( \bar{x}' \) solves \( E_k(\epsilon_j) \) for \( \forall k \)

**Theorem 2.** If \( \bar{x}' \) solves \( E_k(\epsilon_j) \) for \( \forall k \) and \( \bar{x}' \) is a unique solution, then \( \bar{x}' \) is a Pareto optimal solution

The above theorems are proved for the general multi-objective optimization problems (Miettinen, 2012). These theorems indicate that Pareto optimal solutions can always be found by solving \( \epsilon \) – constraint problems, provided \( \epsilon_j \) in \( E_k(\epsilon_j) \) is in the feasible range of PF.

To properly apply the \( \epsilon \) – constraint method we must have the range of every objective function, at least for the \( n - 1 \) objective functions that will be used as constraints. The calculation of the range of the objective functions over the efficient set is not a trivial task. While the best value is easily attainable as the optimal of the individual optimization, the worst value over the efficient set (Nadir value) is not. The most common approach is to calculate these ranges from the payoff table (the table with the results from the individual optimization of the \( n \) objective functions). The nadir value is usually approximated with the minimum of the corresponding column (Berube et al., 2009). However, even in this case, we must be sure that the obtained solutions from the individual optimization of the objective functions are indeed Pareto optimal solutions.
Definition 5: An objective vector $\vec{f}_{NP} = (f_{iNP})$, $i = 1, 2, ..., n$ constructed using the worst value of the objective functions in the complete Pareto front (PF) is called a Naider point.

Let $\vec{f}_{NP} = (f_{iNP})$ be the Nadir point

Where, $f_{iNP} = \max f_i(\vec{x}), \vec{x} \in PS$

Though it looks simple to estimate the Naider point, but it is not that straightforward. To find the Naider point it is required to satisfy the condition that the design variable vector must belong to the Pareto set. The estimation of Naider point from the payoff table can be easily understood from the following example. Consider a multi-objective optimization problem to minimize two functions $f_1$ and $f_2$. We will first minimize $f_1$ which gives the design variable vector as $\vec{x}_1$ and let us suppose the minimized value of $f_1$ = 25. For $\vec{x}_1$ calculate the value of $f_2$, which is say $f_2 = 100$ and say for reverse case (i.e minimizing $f_2$) the values are $f_2 = 15$ and $f_1 = 90$. The payoff table can be constructed as given in Table-1. So, Naider point is $\vec{f}_{NP} = (f_{1NP}, f_{2NP}) = (90, 100)$. The concept can be visualized from the Figure-1, where all the points, Pareto front and Naider point is indicated clearly.

The $\epsilon − HTS$ is explained in Algorithm-2 as follows:

Algorithm 2: $\epsilon − HTS$ algorithm for multi-objective optimization considering two objectives

Decide the number of Pareto solutions $k$ required in the Pareto front (PF)
Minimize each objective function and find corresponding design vector

Find $\vec{x}_1 \rightarrow \min f_1 = f_1^O$ and $\vec{x}_2 \rightarrow \min f_2 = f_2^O$
Calculate Nadir point $(f_{1NP}, f_{2NP}) \rightarrow PF^1 = (f_1^O, f_2^O)$
Set $\epsilon_j = \epsilon_1 = f_1^O + \delta; \delta = \frac{(f_{1NP} - f_1^O)}{k-1}$

While $\epsilon_j \geq f_{1NP}$: Solve $E_k(\epsilon_j)$ through HTS $\rightarrow PF^j = (f_1^O, f_2^O)$
Set $j = j + 1$ and $\epsilon_j = f_1^O + \delta$
if $\exists$ dominated solutions $\rightarrow$ Remove

If the problem, $E_k(\epsilon_j)$, is multimodal, then there are chances to get some dominated solutions due to the presence of many local optima. It is noted from the algorithm provided above that the proposed method is algorithmically simple and do not require special knowledge which is
required or the other multiobjective versions like NSGA, SPEA, PAES etc. If the designer knows how to optimize a single objective problem, it is enough to extend it to develop its multi-objective version by following above Algorithm 2. The choice of effective meta-heuristics overcomes this problem to a great extent, but due to the stochastic nature of different meta-heuristics the global solutions cannot be guaranteed with full confidence. All the meta-heuristics can find global or near global solutions with much higher accuracy compared to the classical techniques, which can only find the local solutions and are highly sensitive to the initial solutions. One of such effective meta-heuristics, heat transfer search (HTS) is considered to find the global solutions to $E_k(\varepsilon_j)$. To build the confidence for the global solutions, it is a good practice to obtain the results for different independent runs with different initial populations. Approximate Pareto front can be obtained by reducing the number of function evaluations than that required to generate true Pareto-front. This can only be achieved if the meta-heuristics possesses good convergence rate. As discussed in (Patel and Savsani, 2005), HTS possesses good convergence rate compared to other state-of-the art algorithms.

5. **Numerical Examples:**

There exist many different test functions for multi-objective optimization (Zhang et al., 2009; Zitzler and Thiele, 1999; Zitzler et al., 2000), but a subset of a few extensively used functions provides a wide range of diverse properties in terms Pareto optimal set and Pareto front. To check the proposed algorithm, we have chosen a subset of these functions with convex, non-convex and discontinuous Pareto fronts. We also include functions with more complex Pareto sets. In this paper, we have tested the following five functions:

- **Schaffer’s Min-Min (SCH) (Schaffer,1985)**
  
  $$f_1(x) = x^2, \quad f_2(x) = (x - 2)^2, \quad -10 \leq x \leq 10$$

- **ZDT1 function (Zitzler, 2000)**
  
  $$f_1(x) = x_1, \quad f_2(x) = g \left(1 - \frac{f_1}{\sqrt{g}}\right)$$

  Where, $g = 1 + \frac{9\sum_{i=2}^{d} x_i}{d - 1}, d = \text{dimension of the problem}$

  $$0 \leq x_i \leq 1, i = 1,2, \ldots 30$$

- **ZDT2 function (Zitzler, 2000)**
  
  $$f_1(x) = x_1, \quad f_2(x) = g \left(1 - \left(\frac{f_1}{g}\right)^2\right)$$
• ZDT3 function (Zitzler, 2000)

\[ f_1(x) = x_1, f_2(x) = g \left( 1 - \frac{f_1}{g} - \frac{f_1}{g} \sin(10\pi f_1) \right) \]

• LZ function (Li and Zhang, 2009)

\[ f_1(x) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left( x_j - \sin \left( 6\pi x_1 + \frac{j\pi}{d} \right) \right)^2 \]

\[ f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left( x_j - \sin \left( 6\pi x_1 + \frac{j\pi}{d} \right) \right)^2 \]

Where, \( J_1 = \{ j | j \text{ is odd} \}, J_2 = \{ j | j \text{ is even} \}, 2 \leq j \leq d. \)

Numerical experiments are carried out for different benchmark multi-objective problems. The benchmark multi-objective functions considered in this work are SCH, ZDT1, ZDT2, ZDT3, and LZ which possess different characteristics. SCH is a one-dimensional problem with convex Pareto front. ZDT1 have convex Pareto front, whereas ZDT2 have non-convex Pareto front. The Pareto front of ZDT3 have discontinuous Pareto front. The value of \( g \) in ZDT2 and ZDT3 are same. The Pareto front for ZDT1, ZDT2 and ZDT3 occurs when value of \( g \) reaches 1. The objective functions in LZ function are multi-modal and it adds additional challenge for the algorithm to find the true Pareto front. The function LZ possesses Pareto front when \( f_2 = 1 - \sqrt{f_1} \). All the problems taken for the study are unconstrained multi-objective functions, but as \( \epsilon - HTS \) works by converting \((n - 1)\) objectives into constraints, the considered problems are indirectly considered as constrained multi-objective problems. There are many methods to handle constrained problems. In this work, all the problems use static penalty method (Rao, 2009) approach. A penalty value is added to the objective function for each infeasible solution so that it will be penalized for violating the constraints. This method is popular because it is simple to apply. It requires amount of penalty to be added and varies for different problems. The constrained optimization problem can be converted in to unconstrained optimization problem using static penalty approach as follows:

\[ f_j(X) = f_j(X) + \sum_{i=1}^{P} P_i \max\{g_i(X), 0\} + \sum_{i=p}^{NC} P_i \max\{|h_i(X)| - \delta, 0\} \]

Where,

\( f_j(X), j = 1, 2 \ldots n, \) are the objective function to be optimized (here minimized)

\( X = \{x_1, x_2, \ldots x_m\}, \) are design variables

\( g_i(X) \leq 0, i = 1, 2 \ldots p \) are inequality constraints
\( h_i(X) = 0, i = p + 1 \ldots NC \) are equality constraints

\( p \) is the total number of nonlinear constraint, \((NC - p)\) is the total number of equality constraints and \( NC \) is total number of constraints. \( P_i \) is a penalty factor which is generally assigned a large number and \( \delta \) is a tolerance on the equality constrained to consider it as a feasible. Here it can be noted that for \( \epsilon - HTS \) additional \((n - 1)\) inequality constraints are added. So, even for unconstrained problem at least \((n - 1)\) inequality constraint exists. Hence, \( \epsilon - HTS \) method can handle constrained multi-objective optimization problems as well.

The performance of the algorithm is calculated by following criteria (Deb, 2001):

- **Generational Distance (GD):**

  GD is calculated between the true Pareto front \((PF^t)\) and the obtained Pareto front \((PF^o)\). GD indicates the average distance between the obtained Pareto front and the true Pareto front. GD is defined as:

  \[
  GD = \frac{\sum_{p=1}^{k} (d_p)}{k}
  \]

  Where, \( d_p = \left( \sum_{i=1}^{n} (PF_{i,p}^o - PF_{i,p}^t)^2 \right)^{1/2} \)

  Where, \( k \) is the number of Pareto Solutions, \( n \) is the number of objective functions, \( PF_{i,p}^o \) indicate the \( p^{th} \) obtained Pareto solution for the \( t^{th} \) objective and \( PF_{i,p}^t \) indicate the nearest point on true Pareto front from \( PF_{i,p}^o \)

- **Spacing (S):**

  Spacing is defined as:

  \[
  S = \frac{1}{k-1} \sum_{p=2}^{k} (D_p - \overline{D})^2
  \]

  Where, \( D_p \) is the absolute difference between the two consecutive solutions in the obtained Pareto front \((PF^o)\). It is defined as:

  \[
  D_p = \sum_{i=1}^{n} |PF_{i,(p-1)}^o - PF_{i,p}^o|
  \]

  \( \overline{D} \) is the average of all \( D_p \).

Spacing(S) specify the spread of the obtained Pareto front. It gives the standard deviation of \( D_g \). The small value of S indicates the uniform spacing of the obtained Pareto Solutions.
• **Spread** ($\Delta$):

Spread is defined as:

$$\Delta = \frac{\sum_{j=1}^{n} d_{j}^{ex} + \sum_{p=2}^{k} |d_{p} - \bar{d}|}{\sum_{j=1}^{n} d_{j}^{ex} + (k - 1)\bar{d}}$$

Where, $d_{p}$ is the Euclidian distance between two consecutive points of the obtained Pareto front and $d_{j}^{ex}$ is the Euclidian distance between the obtained Pareto front and the True Pareto front. $\bar{d}$ is the average of $d_{p}$. Spread checks the condition for the obtained Pareto front to cover the True Pareto front. Smaller value of $\Delta$ indicate better spread for the obtained Pareto front with uniform distribution.

The $\epsilon$–HTS requires specific parameters for its execution, which includes CDF, COF and RDF. These parameters are considered as given in Patel and Savsani (2015), i.e CDF=RDF=2 and COF=10 as these parameter setting has performed better on different benchmark functions. For the experimentation, the population size is set as 20 and maximum function evaluations as 10000 for ZDT1, ZDT2, ZDT3 and SCH functions. For LZ function the population size is set as 50 and function evaluations as 300000, to maintain the uniformity of experimental setup. For all the problems number of Pareto Solutions ($\epsilon$) are considered as 200.

The Pareto front for SCH, ZDT1, ZDT2, ZDT3 and LZ are show in Figure 2 to 6 respectively. All the figures show the True Pareto front and the obtained Pareto front. It can be noted form the results that for all the functions $\epsilon$–HTS is capable to approximate the True Pareto front. The effectiveness of the results is checked by the above-mentioned performances measures. The results for the GD for all the functions are summarized in Table 2, where the results are compared with the state-of-the art multi-objective algorithms such as NSGA-2, SPEA, VEGA, MODE, DEMO, MO-Bees, MOFA and MOCS. The results except $\epsilon$–HTS are represented from Yang, 2013 and Yang and Deb, 2013. The results summarized in Table 2 are the average result obtained in 25 independent runs.

It is observed from the results that $\epsilon$–HTS has produced better GD compared to all the algorithms except for ZDT3 for which the results are only inferior to MOCS. It can be summarized that $\epsilon$–HTS can find true Pareto solutions for the problems with convex, non-convex and discrete Pareto front. The results are also compared based on the spread ($\Delta$) with the other multi-objective optimization algorithms such as NSGA-II (real coded), NSGA-II(binary coded), SPEA and PAES. The result for the $\Delta$ are shown in Table 3. It can be noted form the results that $\epsilon$–HTS has shown better $\Delta$ for ZDT1 compared to all the algorithms. For, ZDT2, $\epsilon$–HTS is better compared to SPEA and PAES, but slightly inferior to NSGA-II. For ZDT3 value of $\Delta$ for $\epsilon$–HTS is inferior to all the algorithms.
6. Engineering design experiments:

In this section $\epsilon - HTS$ is experimented on 6 different multi-objective engineering design problems which are widely used in the research to investigate the multi-objective optimization algorithms. All the considered problems possess different characteristics. The algorithms parameters for the $\epsilon - HTS$ is same as that for the benchmark problems considered in the previous section. All the engineering problems are experimented with a population size of 20 and 20000 function evaluations. For all the problems 200 Pareto solutions are generated and the results mentioned in this work are the average result obtained in the 30 runs to maintain the same experimental set up with (Sadollah et al., 2015).

- **Four bar truss problem:**

  The aim of this problem is to minimize volume and displacement of joints simultaneously. Area of each link is considered as a design variables. This problem is an unconstrained problem with all the continuous design variables. The system is shown in Figure 7.

  The problem can be stated as below

  \[ Minimize: f_1(x) = L \left( 2x_1 + \sqrt{2x_2 + \sqrt{x_3}} + x_4 \right) \]

  \[ Minimize: f_2(x) = \left( \frac{F}{E} \right) \left( \frac{2}{x_2} + 2 \frac{\sqrt{2}}{x_2} - 2 \frac{\sqrt{2}}{x_3} + 2 \frac{\sqrt{2}}{x_4} \right) \]

  Where,

  \[ F = 10, \quad E = 2e5, \quad L = 200 \]

  \[ 1 \leq x_1, x_4 \leq 3, \quad \sqrt{2} \leq x_2, x_3 \leq 3 \]

  This problem has a convex Pareto front in the objective space. The Naider point for this problem is obtained as (1727.739, 0.03299) and the extreme points of the Pareto front obtained by using $\epsilon - HTS$ for this problem are (1180, 0.032994) and (1727.7394, 0.0027614). Mathematical expression for the exact Pareto front is not available for this problem in the literature, so the performance is checked based on the value of spacing (S) due to its availability in the literature for different algorithms. The value of S for $\epsilon - HTS$ and different algorithms such as NSGA-II, MOPSO, Micro-GA, PAES and MOWCA (Sadollah et al., 2015) are given in Table 4. It can be observed from the results that $\epsilon - HTS$ has produced minimum value of spacing (S) with reasonable standard deviation (SD) among the solutions compared to the other algorithms. Hence it can be interpreted that $\epsilon - HTS$ has produced better spacing among other comparative algorithms. The better spacing indicates that the Pareto solutions are uniformly distributed along the Pareto front. The obtained Pareto front is presented in Figure 8, which justifies the results presented in the Table 4 and also the extreme points of the Pareto front can be visualized that coincide with the Naider point.
obtained Pareto front matches well with the available Pareto front in the literature (Sadollah et al., 2015)

- **Gear train Problem:**

The purpose of this problem is to minimize the maximum size of any one of the gear simultaneously with the minimization of the gear ratio error with reference gear ratio of $1/6.931$. All the design variables considered in this example can only take integer values as it represents the number of teeth on each gear. Hence, this problem is a min-max problem with discrete design variables, which offers additional challenges to an optimization algorithm. The system is shown in Figure 9.

The problem can be stated as:

$$Minimize \ f_1(x) = \left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right)^2$$

$$Minimize \ f_2(x) = \max([x_1 \ x_2 \ x_3 \ x_4])$$

Where, $12 \leq x_1, x_2, x_3, x_4 \leq 60$

The performance measures such as GD, $S$ and $\Delta$ are not available for the gear train problems. So, the results are compared based on the extreme Pareto solutions available in the literature. The results are compared with NSGA-II and MOWCA. The results are summarized in Table 5.

It can be observed from the results that the extreme Pareto solutions obtained by $\epsilon - HTS$ is $(9.92e-10, 47)$ and $(5.01e-1, 13)$. The value of $f_1$ for the upper left corner of Pareto front obtained by $\epsilon - HTS$ is $9.92e-10$ which is lower than $1.83e-8$ produced by NSGA-II and $4.5e-9$ produced by MOWCA. So, it can be interpreted that the upper extreme result of Pareto front produced by $\epsilon - HTS$ is better compared to both the algorithms. The value of $f_2$ at lower right corner of Pareto front obtained by $\epsilon - HTS$ is $13$ which is same that of NSGA-II and slightly higher to MOWCA (Sadollah et al., 2015). So, $\epsilon - HTS$ has produced slight inferior results compared to MOWCA and equivalent to NSGAIi for $f_2$. The Pareto front for this problem is plotted on a log-scale in figure 10 to visualize the results properly. The errors produced on the Pareto front can be easily understood and interpreted from the log-scale. Also, several discrete points can be observed on the extreme left part of the Pareto front which indicates the lower level of tooth meshing errors. Each point of the extreme left of the Pareto front produced indicates the tooth meshing errors and its variation on the log scale. Effective algorithms are required to find such points. This problem is a type of combinatorial problem and combination of design variables to produce such extreme results is highly desirable by design engineers.
Multi-plate disc brake design:

Multi-plate disc brake finds its applications in airplanes, to apply effective braking while landing. The exploded view of the multi-pate disc brake is shown in figure 11. The purpose of the problem is to simultaneously minimize the mass of the brake and the stopping time. There are four design variables for the inner radius, outer radius, the engaging force (applied force) and the number of friction surfaces (number of friction plates). Out of this four-design variable number of friction surfaces can only take a discrete integer values which make it a mixed-integer problem. This is also a constrained problem for which five different restrictions are introduced for the distance between the radii of the friction plates, length of the brake, pressure sustained by the plates, maximum limitation for the temperature generated and the braking torque. The problem can be stated as follows:

Minimize \( f_1(x) = 4.9e - 5(x_2^2 - x_1^2)(x_4 - 1) \)

Minimize \( f_2(x) = (9.82e6) \frac{x_2^2 - x_1^2}{x_3x_4(x_2^3 - x_1^3)} \)

\( g_1 = 20 + x_1 - x_2 \)

\( g_2 = 2.5(x_4 + 1) - 30 \)

\( g_3 = \frac{x_3}{3.14(x_2^2 - x_1^2)^2} - 0.4 \)

\( g_4 = 2.22e - 3x_3 \frac{x_3^3 - x_1^3}{(x_2^2 - x_1^2)^2} - 1 \)

\( g_5 = 900 - \left( \frac{2.66e - 2x_3x_4(x_3^3 - x_1^2)}{x_2^2 - x_1^2} \right) \)

\( \forall g \leq 0 \)

\( 55 \leq x_1 \leq 80, \quad 75 \leq x_2 \leq 110, \quad 1000 \leq x_3 \leq 3000, \quad 2 \leq x_4 \leq 20, \quad x_4 \in \mathbb{I} \)

The Naider point used by \( \epsilon - HTS \) for this problem is \((2.793, 16.6909)\) and the extreme Pareto solutions obtained are \((0.1634, 16.6909)\) and \((2.793, 2.071)\). The results are compared for the value of spacing (\( \Delta \)). The results are shown in Table 6, where the performance of \( \epsilon - HTS \) is compared with NSGA-II, \( pae - ODEMO \) and MOWCA. It can be noted from the results that \( \epsilon - HTS \) has produced better \( \Delta \) compared to NSGA-II, and \( pae - ODEMO \). The performance of \( \epsilon - HTS \) is slight inferior compared to MOWCA. The Pareto front obtained by using \( \epsilon - HTS \) is shown in Figure 12, which is either same or better than the Pareto front given in (Sadollah et al., 2015). It is useful for the designer to note that the Pareto solutions becomes nearly stagnant for \( f_1 \) at \( f_1 \simeq 0.2 \) and for \( f_2 \) at \( f_2 \simeq 1.75 \).
• Speed reducer problem:

The aim of this problem is to simultaneously optimize the weight of the gear assembly and the transverse deflection of the shaft. The assembly of the speed reducer is shown in Figure 13. The problem is subjected to different constraints on bending stress of the gear teeth, surfaces stress, transverse deflections of the shafts and stresses in the shafts. The design variables are the face width, module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts respectively. All the variables are indicated in the figure 13. The third variable is integer, the rest of them are continuous. So, this problem can be classified as a mixed-integer problem.

The problem can be stated as follows:

**Minimize** $f_1$

$$f_1 = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

**Minimize** $f_2 = \frac{\sqrt{(745 \frac{x_4}{x_2x_3})^2 + 1.69e7}}{0.1x_6^3}$

$g_1 = -(27/(x_1x_2^2x_3) - 1)$

$g_2 = -(397.5/(x_1x_2^2x_3^3) - 1)$

$g_3 = -\left(1.93 - \frac{x_4^3}{x_2x_3x_6^4} - 1\right)$

$g_4 = -\left(1.93 - \frac{x_5^3}{x_2x_3x_7^4} - 1\right)$

$g_5 = -\left(\frac{\sqrt{(745 \frac{x_4}{x_2x_3})^2 + 16.9e6}}{0.1x_6^3} - 1\right)$

$g_6 = -\left(\frac{\sqrt{(745 \frac{x_5}{x_2x_3})^2 + 157.5e6}}{0.1x_7^3} - 1\right)$

$g_7 = -(x_2 \times \frac{x_3}{40} - 1)$
\[ g_8 = -\left( 5 \frac{x_2}{x_1} - 1 \right) \]

\[ g_9 = -\left( \frac{x_1}{12 \times x_2} - 1 \right) \]

\[ g_{10} = -\left( \frac{1.5 \times x_6 + 1.9}{x_4} - 1 \right) \]

\[ g_{11} = -\left( \frac{1.1 \times x_7 + 1.9}{x_5} - 1 \right) \]

\[ \forall g_1 \geq 0 \]

\[ 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4, x_5 \leq 8.3, \]

\[ 2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5 \]

Naider point used by \( \epsilon - HTS \) for this problem is \((5969.91, 1071.54)\). The extreme Pareto solution obtained by \( \epsilon - HTS \) is \((3002.11, 1071.54)\) and \((5969.91, 694.71)\). The results are compared based on the Spacing (S) with the other algorithms such as NSGA-II, Micro-GA, PAES and MOWCA (Sadollah et al., 2015). The results are presented in the Table7.

It can be observed from the results that, the Spacing of NSGA-II better compared to all the algorithms, but the extreme points of the Pareto front obtained by NSGA-II is inferior compared to other algorithms. Also, value of S is better for \( \epsilon - HTS \). The Pareto front is given in Figure 14. It is useful for the designer to note that the Pareto solutions becomes nearly stagnant for \( f_2 \) at \( f_2 \approx 697 \).

- **Welded beam design:**

The purpose of this problem is to minimize the cost simultaneously with the end deflection. The problem is subjected to the constraints on shear stress, bending stress, weld length and the buckling load. The four different design variables are the height and the length of the welded joint and thickness and the width of the beam. All the design variables are continuous. The schematic diagram is shown in figure 15, where all the variables are shown clearly. The value of objective functions differs much in its values and so such problems are difficult to solve with the weighted sum method. The problem can be stated as:

\[ \text{Minimize } f_1 = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2) \]

\[ \text{Minimize } f_2 = \text{del} \]

\[ g_1 = -(tau - taumax) \]

\[ g_2 = -(sig - sigmax) \]
\( g_3 = -(x_1 - x_4) \)
\( g_4 = -(P - pc) \)

Where,

\[
pc = \left( \frac{4.013 \times E \left( \frac{x_3^2 + x_4^2}{36} \right)}{L^2} \right) \left( 1 - \left( \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right) \right)
\]

\[
del = \frac{4PL^3}{Ex_3 x_4}
\]

\[
sig = \frac{6PL}{x_4 x_3^2}
\]

\[
J = 2 \times \left( \sqrt{2x_1 x_2} \left( \left( \frac{x_2^2}{12} \right) + \left( \frac{x_1 + x_3}{2} \right)^2 \right) \right)
\]

\[
R = \sqrt{\left( \frac{x_2^2}{4} \right) + \left( \frac{x_1 + x_3}{2} \right)^2}
\]

\[
M1 = P \times \left( L + \frac{x_2}{2} \right)
\]

\[
tau2 = \frac{M1R}{J}
\]

\[
tau1 = \frac{P}{\sqrt{2x_1 x_2}}
\]

\[
tau = \sqrt{\tau_1^2 + 2 \cdot \tau_1 \cdot \tau_2 \cdot \frac{x_2}{2 \cdot R} + \tau_2^2}
\]

\( \forall g \geq 0 \)

\[ P = 6000, L = 14, E = 30e6, G = 12e6, taumax = 13600, sigmax = 30000, \]

\[ delmax = 0.25 \]

\[ 0.125 \leq x_1, x_4 \leq 5, 0.1 \leq x_2, x_3 \leq 10 \]
The Naider point for this problem is taken as (35.3476, 0.014452). the extreme Pareto solutions obtained by $\epsilon - HTS$ are (1.725, 0.014452) and (35.3476, 0.00043904). This problem is compared with the other techniques such as NSGA-II, paε - ODEMO, and MOWCA (Sadollah et al., 2015). The results are compared based on $\Delta$ and the mean values along with the calculated standard deviation are summarized in Table 8. It can be observed from the results that $\epsilon - HTS$ possesses better spreading compared to the other algorithms. The Pareto front obtained by $\epsilon - HTS$ is shown in Figure 16, where it justifies the results.

- **Spring design problem:**

The purpose of this problem is to minimize stress and volume simultaneously. The constraints are imposed on the minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the wire diameter ($d$), the mean coil diameter ($D$), and the number of active coils ($N$). The schematic view of the spring is shown in Figure 17. This problem is special because all the design variables possess different characteristics. The number of turns can only take integer values where as diameter of the wire is standardized and it has to be selected from the set of available diameters. The mean coil diameter can be considered as a continuous variable. So, this problem is mixed-integer-discrete problem. The problem can be stated as follows:

$$Minimize \ f_1(x) = (0.253.14^2x_2^2x_3(x_1 + 2))$$

$$Minimize \ f_2(x) = 8KP_{\text{max}} \frac{x_3}{3.14x_2^3}$$

$$g_1 = 1.05x_2(x_1 + 2) + \frac{p_{\text{max}}}{k} - l_{\text{max}}$$

$$g_2 = d_{\text{min}} - x_2$$

$$g_3 = x_2 + x_3 - D_{\text{max}}$$

$$g_4 = 3 - C$$

$$g_5 = d_{\text{el}} - d_{\text{el}_{\text{pm}}}$$

$$g_6 = d_{\text{el}} - \frac{P_{\text{max}} - P}{k}$$

$$g_7 = 8KP_{\text{max}} \frac{x_3}{3.14x_2^3} - S$$

$$g_8 = (0.253.14^2x_2^2x_3(x_1 + 2)) - V_{\text{max}}$$

Where,
\[
P = 300
\]
\[
D_{\text{max}} = 3
\]
\[
V_{\text{max}} = 30
\]
\[
P_{\text{max}} = 1000
\]
\[
del_w = 1.25
\]
\[
l_{\text{max}} = 14
\]
\[
del_{pm} = 6
\]
\[
d_{\text{min}} = 0.2
\]
\[
S = 189000
\]
\[
G = 11500000
\]
\[
C = \frac{x_3}{x_2}
\]
\[
k = G \frac{x_2^4}{8x_1x_3^2}
\]
\[
del_p = \frac{p}{k}
\]
\[
K = \frac{4C - 1}{4C - 4} + \left(0.615 \frac{x_2}{x_3}\right)
\]
\[
\forall g \leq 0
\]
\[
1 \leq x_1 \leq 32, 1 \leq x_3 \leq 30, x_1 \in I
\]
\[
x_2 \in \{0.009, 0.0095, 0.0104, 0.0118, 0.0128, 0.0132, 0.014, 0.015, 0.0162, 0.0173, 0.018, 0.020, 0.023, 0.025, 0.028, 0.032, 0.035, 0.041, 0.047, 0.054, 0.063, 0.072, 0.080, 0.092, 0.105, 0.120, 0.135, 0.148, 0.162, 0.177, 0.192, 0.207, 0.225, 0.244, 0.263, 0.283, 0.307, 0.331, 0.362, 0.394, 0.4375, 0.5\}
\]

This value for GD, S or \(\Delta\) is not available in the literature for this problem so the results are compared with the extreme Pareto solutions. The extreme Pareto solutions are compared with NSGA-II and MOWCA. The results are presented in the Table 9. It can be observed from the results that the left extreme point obtained by \(\epsilon - HTS\) is slight inferior to NSGA-II and MOWCA (Sadollah et al., 2015), but \(\epsilon - HTS\) has obtained better non-dominated solutions
on the right extreme. The Pareto front is shown in Figure 18, which justifies the results. The Pareto front of this problem is discontinuous and of overlapping nature. This occurs because each discrete value for $d$ possesses certain portion of the Pareto front, which can be observed from the Pareto front for different discrete values of $d$.

7. Conclusions:

Multi-objective version for heat transfer search (HTS) algorithm, called $\epsilon - HTS$, is proposed in this work. The proposed algorithm is experimented on numerical benchmark problems with different characteristics of Pareto front. The proposed method is also applied to different engineering design problems. The performance of the algorithm is calculated based on generational distance, spacing and spreading along with its Pareto front. The results of $\epsilon - HTS$ are compared with the different multi-objective variants of GA, PSO, DE, Bees and WCA. The results indicate that the proposed method is better or as par with the existing algorithms for solving multi-objective problems.

Acknowledgments:

The research of the 2nd author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The postdoctoral fellowship of the 1st author is supported by NSERC.

Conflict of Interest:

The author has confirmed that there is no conflict of interest.

References:


Table 1 Method to construct Payoff-Table

<table>
<thead>
<tr>
<th>Single objective optimization</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>Naider point</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1 \rightarrow \min$</td>
<td>25</td>
<td>100</td>
<td>$f_2^{NP}$ 100</td>
</tr>
<tr>
<td>$f_2 \rightarrow \min$</td>
<td>90</td>
<td>15</td>
<td>$f_1^{NP}$ 90</td>
</tr>
</tbody>
</table>

Table 2: Results(GD) for benchmark multi-objective problems

<table>
<thead>
<tr>
<th></th>
<th>SCH</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>LZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>VEGA</td>
<td>6.98E-02</td>
<td>3.79E-02</td>
<td>2.37E-03</td>
<td>3.29E-01</td>
<td>1.47E-03</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>5.73E-03</td>
<td>3.33E-02</td>
<td>7.24E-02</td>
<td>1.14E-01</td>
<td>2.77E-02</td>
</tr>
<tr>
<td>MODE</td>
<td>9.32E-04</td>
<td>5.80E-03</td>
<td>5.50E-03</td>
<td>2.15E-02</td>
<td>3.19E-03</td>
</tr>
<tr>
<td>DEMO</td>
<td>1.79E-04</td>
<td>1.08E-03</td>
<td>7.55E-04</td>
<td>1.18E-03</td>
<td>1.40E-03</td>
</tr>
<tr>
<td>Bees</td>
<td>1.25E-02</td>
<td>2.40E-02</td>
<td>1.69E-02</td>
<td>1.91E-01</td>
<td>1.88E-02</td>
</tr>
<tr>
<td>SPEA</td>
<td>5.17E-03</td>
<td>1.78E-03</td>
<td>1.34E-03</td>
<td>4.75E-02</td>
<td>1.92E-03</td>
</tr>
<tr>
<td>MOCS</td>
<td>1.25E-07</td>
<td>1.17E-07</td>
<td>2.23E-05</td>
<td>2.88E-05</td>
<td>4.19E-05</td>
</tr>
<tr>
<td>MOFA</td>
<td>4.55E-06</td>
<td>1.90E-04</td>
<td>1.52E-04</td>
<td>1.97E-04</td>
<td>8.70E-04</td>
</tr>
<tr>
<td>$\varepsilon \rightarrow HTS$</td>
<td>3.02E-09</td>
<td>7.01E-10</td>
<td>1.40E-11</td>
<td>1.32E-04</td>
<td>6.22E-09</td>
</tr>
</tbody>
</table>
Table 3 Results ($\Delta$) for the multi-objective benchmark problems

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II(RC)</td>
<td>0.3903</td>
<td>0.4307</td>
<td>0.7385</td>
</tr>
<tr>
<td>NSGA-II(BC)</td>
<td>0.4632</td>
<td>0.4351</td>
<td>0.5756</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.7301</td>
<td>0.6781</td>
<td>0.6657</td>
</tr>
<tr>
<td>PAES</td>
<td>1.2297</td>
<td>1.1659</td>
<td>0.78992</td>
</tr>
<tr>
<td>$\epsilon - HTS$</td>
<td>0.3359</td>
<td>0.4539</td>
<td>1.2741</td>
</tr>
</tbody>
</table>

Table 4 Result ($S$) for the four-bar truss problem

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$S$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>NSGA-II</td>
<td>2.3635</td>
<td>0.2551</td>
</tr>
<tr>
<td>MOPSO</td>
<td>2.5303</td>
<td>0.2275</td>
</tr>
<tr>
<td>Micro-GA</td>
<td>8.2742</td>
<td>16.8311</td>
</tr>
<tr>
<td>PAES</td>
<td>3.2314</td>
<td>5.9555</td>
</tr>
<tr>
<td>MOWCA</td>
<td>2.5816</td>
<td>0.0298</td>
</tr>
<tr>
<td>$\epsilon - HTS$</td>
<td>1.1053</td>
<td>0.2352</td>
</tr>
</tbody>
</table>

Table 5: Result (extreme Pareto solutions) for the gear train problem

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Objective function</th>
<th>$f_1$</th>
<th>$f_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>$f_1 \rightarrow min$</td>
<td>1.83e-8</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>$f_2 \rightarrow min$</td>
<td>5.01e-1</td>
<td>13</td>
</tr>
<tr>
<td>MOWCA</td>
<td>$f_1 \rightarrow min$</td>
<td>4.5e-9</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>$f_2 \rightarrow min$</td>
<td>7.32e-1</td>
<td>12</td>
</tr>
<tr>
<td>$\epsilon - HTS$</td>
<td>$f_1 \rightarrow min$</td>
<td>9.92e-10</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>$f_2 \rightarrow min$</td>
<td>5.01e-1</td>
<td>13</td>
</tr>
</tbody>
</table>
Table 6 Result (Δ) for the disc brake problem

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>∆</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.79717</td>
<td>0.06608</td>
</tr>
<tr>
<td>paε − ODEMO</td>
<td>0.8401</td>
<td>0.20085</td>
</tr>
<tr>
<td>MOWCA</td>
<td>0.46041</td>
<td>0.10961</td>
</tr>
<tr>
<td>ε − HTS</td>
<td>0.6991</td>
<td>0.09257</td>
</tr>
</tbody>
</table>

Table 7 Result (S) for speed reducer problem

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>S</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSGA-II</td>
<td>2.765</td>
<td>3.534</td>
</tr>
<tr>
<td>Micro-GA</td>
<td>47.80</td>
<td>32.80</td>
</tr>
<tr>
<td>PAES</td>
<td>16.20</td>
<td>4.268</td>
</tr>
<tr>
<td>MOWCA</td>
<td>16.68</td>
<td>2.697</td>
</tr>
<tr>
<td>ε − HTS</td>
<td>14.38</td>
<td>1.864</td>
</tr>
</tbody>
</table>

Table 8 Results (Δ) for welded beam problem

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>∆</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSGA-II</td>
<td>0.8898</td>
<td>0.1197</td>
</tr>
<tr>
<td>paε − ODEMO</td>
<td>0.5860</td>
<td>0.0436</td>
</tr>
<tr>
<td>MOWCA</td>
<td>0.2247</td>
<td>0.0928</td>
</tr>
<tr>
<td>ε − HTS</td>
<td>0.2032</td>
<td>0.1024</td>
</tr>
</tbody>
</table>

Table 9 Result (extreme Pareto solutions) for spring problem

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>f₁ → min</th>
<th>f₂ → min</th>
<th>f₁</th>
<th>f₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSGA-II</td>
<td>2.690</td>
<td>187053</td>
<td>24.189</td>
<td>61949</td>
</tr>
<tr>
<td>MOWCA</td>
<td>2.668</td>
<td>188448</td>
<td>26.93</td>
<td>58752</td>
</tr>
<tr>
<td>ε − HTS</td>
<td>2.8241</td>
<td>183679.6</td>
<td>28.0502</td>
<td>56833.06</td>
</tr>
</tbody>
</table>
Figure 1 Placement of Naider Point

Naider Point
(90,100) =
\( (f_1^{NP}, f_2^{NP}) \)
Figure 2 Pareto front for SCH function
Figure 3 Pareto front for ZDT1 function
Figure 4 Pareto front for ZDT2 function
Figure 5 Pareto front for ZDT3 function
Figure 6 Pareto front for LZ function

Figure 6 Pareto front for LZ function
Figure 7: Four bar truss problem
Figure 8 Pareto front for the four-bar truss problem
Figure 9 Gear train
Figure 10 Pareto front for gear train problem
Figure 11 Exploded view for the multi-plate disc brake
Figure 12  Pareto front for disc brake problem
Figure 13 Speed reducer

Figure 14 Pareto front for speed reducer problem
Figure 15 Welded beam
Figure 16 Pareto front for welded beam problem
Figure 17 Spring
Figure 18 Pareto front for spring problem
Highlights:

- A novel multi-objective optimization (MOO) algorithm is proposed.
- Proposed algorithm is presented to obtain the Pareto-optimal solutions.
- The multi-objective optimization algorithm compared with other work in the literature.
- Test performance of proposed algorithm on MOO benchmark/design engineering problems.