



Research article

Applying GA-PSO-TLBO approach to engineering optimization problems

YoungSu Yun^{1,*}, Mitsuo Gen² and Tserengotov Nomin Erdene¹

¹ School of Business Administration, Chosun University, 309 Pilmun-daero, Dong-gu, Gwangju 61452, Republic of Korea

² Fuzzy Logic Systems Institute, Fukuoka and Tokyo University of Science, Tokyo, Japan

* **Correspondence:** Email: ysyun@chosun.ac.kr; Tel: +82622306243; Fax: +82622085208.

Abstract: Under addressing global competition, manufacturing companies strive to produce better and cheaper products more quickly. For a complex production system, the design problem is intrinsically a daunting optimization task often involving multiple disciplines, nonlinear mathematical model, and computation-intensive processes during manufacturing process. Here is a reason to develop a high performance algorithm for finding an optimal solution to the engineering design and/or optimization problems. In this paper, a hybrid metaheuristic approach is proposed for solving engineering optimization problems. A genetic algorithm (GA), particle swarm optimization (PSO), and teaching and learning-based optimization (TLBO), called the GA-PSO-TLBO approach, is used and demonstrated for the proposed hybrid metaheuristic approach. Since each approach has its strengths and weaknesses, the GA-PSO-TLBO approach provides an optimal strategy that maintains the strengths as well as mitigates the weaknesses, as needed. The performance of the GA-PSO-TLBO approach is compared with those of conventional approaches such as single metaheuristic approaches (GA, PSO and TLBO) and hybrid metaheuristic approaches (GA-PSO and GA-TLBO) using various types of engineering optimization problems. An additional analysis for reinforcing the performance of the GA-PSO-TLBO approach was also carried out. Experimental results proved that the GA-PSO-TLBO approach outperforms conventional competing approaches and demonstrates high flexibility and efficiency.

Keywords: hybrid metaheuristic approach; genetic algorithm; particle swarm optimization; teaching and learning-based optimization; engineering optimization problem

1. Introduction

Under addressing global competition, manufacturing companies strive to produce better and cheaper products more quickly. For a complex system such as an aircraft, the design problem is intrinsically a daunting optimization task often involving multiple disciplines, nonlinear mathematical model, and computation-intensive processes during manufacturing process. Here is a reason to develop a high performance algorithm based on soft computing technique for finding an optimal solution to the engineering design and/or optimization problems [1,2]. An engineering optimization problem can be defined as a problem to find an optimal solution while satisfying complex constraints that consist of various types of design variables, including integers, 0–1 variables, and discrete and continuous variables. Since the types of design variables are varied and they are used together in constraints, various approaches to efficiently manage them have been developed in prior research [3–7].

Amir and Hasegawa [3] *suggested* an approach to solve the reinforced concrete beam design problem with three design variables that consisted of an integer variable, a continuous variable, and a predefined discrete variable. To handle three design variables, they converted the continuous and integer variables into discrete ones using the modified Rosenbrocks orthogonalization procedure. However, this led to the loss of some decimal values of the continuous variable in the process of converting the variable types. Therefore, the best solution obtained using this process may not be the optimal solution. Sandgran [4] suggested a branch and bound (B&B) method to solve the gear train and pressure vessel design problems, wherein discrete, continuous and integer variables are used together in constraints. To apply the B&B method to the gear train and pressure vessel design problem, he converted the discrete and integer variables into continuous variables. However, during this conversion process, the search ranges of discrete and integer variables were also changed, which may have led to the B&B method finding a feasible solution, but not an optimal one.

Similar to Sandgran [4], Fu et al. [5] also considered the gear train and pressure vessel design problems. They suggested the integer-discrete-continuous nonlinear programming (IDCNLP) method that changes the type of variables by introducing a penalty function to the discrete and integer variables. However, in the process of introducing a penalty function, the original search space of the design variables may be changed, which may cause the IDCNLP method to not locate the optimal solution or attain premature convergence by getting stuck at a local optimum.

The conventional studies mentioned above usually either converted the type of design variables or introduced a penalty function to the original design variables. However, during conversion or penalty function introduction, either the quality of the obtained solution deteriorates or a premature convergence occurs. To mitigate this situation, various metaheuristic approaches have been developed. A major advantage of applying metaheuristic approaches to engineering optimization problems is that the original design variable types can be maintained without any conversion process [8–12]. Some conventional studies that have implemented such approaches are noted below.

Wu and Chow [9] suggested a genetic algorithm (GA) approach to solve a coil compression spring design problem with a mixture of discrete, continuous and integer design variables. They proved the efficiency of the GA approach by comparing it with the conventional approaches suggested by Amir and Hasegawa [3], Sandgran [4], and Fu et al. [5]. The efficiency of applying the GA approach to various engineering optimization problems was also demonstrated by Lin and Hajela [8] and Yokota et al. [10]. Gandomi et al. [11] solved a corrugated bulkhead design problem and three-bar truss design problem using a cuckoo search (CS) approach. In a recent study, Maputi and Arora [12] suggested a

teaching and learning-based optimization (TLBO) approach. As a repeating and searching approach, the TLBO can guide the search direction to an optimal solution through discussing and learning processes between a teacher and their students. The TLBO approach was applied to a weight gear design problem and a gear transmission system design problem. They proved that applying a TLBO approach outperformed the GA, simulated annealing (SA), and particle swarm optimization (PSO) approaches.

Since most of the conventional studies using various metaheuristic approaches mentioned above can handle the type of original design variables without any conversion process, they have the merit in solving various engineering optimizations when compared with Amir and Hasegawa [3], Sandgran [4], Fu et al. [5], Kuo et al. [6], and Yarushkina [7].

However, in recent years, several studies have shown that a hybrid metaheuristic approach that combines the merits of multiple metaheuristic approaches is more efficient than applying a single metaheuristic approach such as GA, SA, PSO, TLBO, etc. [13–20]. For instance, while GA approach has high performance in global searchability, it has low performance in local searchability. However, SA and hill climbing (HC) approaches have strong local searchability. Therefore, a hybrid approach using GA and SA (or HC) approaches is more efficient than the individual GA, SA, or HC approaches.

In this paper, a hybrid metaheuristic approach is proposed and demonstrated to efficiently solve various engineering optimization problems. The proposed hybrid metaheuristic approach, called the GA-PSO-TLBO approach, combines the GA approach's global searchability, PSO approach's local searchability, and TLBO approach's learning ability. The performance of the GA-PSO-TLBO approach is compared with those of conventional single and hybrid metaheuristic approaches using various engineering optimization problems through a case study. The main contributions of this paper when compared with conventional studies are as follows:

- It presents an efficient hybrid metaheuristic approach by combining the GA approach's global searchability, PSO approach's local searchability, and TLBO approach's learning ability.
- It proves the efficiency of the GA-PSO-TLBO approach by comparing it with conventional single and hybrid metaheuristic approaches.
- It shows the robustness of the GA-PSO-TLBO approach through additional analyses using changes in the number of populations at the initial stage and that of the offspring at the TLBO search process.

This paper is organized as follows: Section 2 describes the definition and characteristics of engineering optimization problems. The detailed implementation procedure of the GA-PSO-TLBO approach is described in Section 3. Section 4 shows the computational results and analyses using various engineering optimization problems to prove the efficiency of the GA-PSO-TLBO approach compared to some conventional single and hybrid metaheuristic approaches, and also includes an additional analysis for providing the robustness of the GA-PSO-TLBO approach. Finally, Section 5 concludes for this paper and suggests a potential future research direction.

2. Mathematical model for engineering optimization problem

Considering minimization problem, the general structure of an engineering optimization problem is as follows [21]:

$$\text{minimize } f(\mathbf{x}) \quad (1)$$

$$\text{subject to } g_i(\mathbf{x}) = 0, i = 1, \dots, k \quad (2)$$

$$g_i(x) \geq 0, i = k + 1, \dots, I \quad (3)$$

$$x_j^L \leq x_j \leq x_j^U, j = 1, \dots, J \quad (4)$$

where f : a scalar objective; $\mathbf{x} = \{x_1, \dots, x_J\}^T = \{x^D, x^I, x^A, x^C\}^T$; \mathbf{x}_j^L : set of lower bounds on design variables x_j ; \mathbf{x}_j^U : set of upper bounds on design variables x_j ; $\mathbf{x}^D \in R^D$: feasible subset of discrete variables; $\mathbf{x}^I \in R^I$: feasible subset of integer variables; $\mathbf{x}^A \in R^A$: feasible subset of 0–1 variables; and $\mathbf{x}^C \in R^C$: feasible subset of continuous variables.

Equation (1) represents the minimization of the design or manufacturing cost under consideration in the engineering optimization problem, which is generally represented as a linear or nonlinear function type. If Eq (1) is represented as a nonlinear function type with a mixture of discrete, continuous, integer, and 0–1 design variables, the conventional approach dictates that all design variables be converted into a single variable type to locate the optimal solution [3–7]. However, as mentioned previously, this may cause either the quality of the obtained solution to deteriorate or premature convergence to occur. Eqs (2)–(4) represent the constraints and each design variables type for solving the minimization problem of Eq (1).

3. Proposed GA-PSO-TLBO approach

As mentioned in Section 1, previous research has shown that applying a hybrid metaheuristic approach that combines the merits of multiple metaheuristic approaches is more efficient than applying those metaheuristic approaches individually [13–19,22]. This paper proposes the GA-PSO-TLBO approach as a hybrid metaheuristic approach that aims to efficiently solve engineering optimization problems with various types of design variables. Since the GA-PSO-TLBO approach has the merits of each metaheuristic approach (GA, PSO and TLBO) in its search processes, effectively combining their merits is an important problem.

First, the characteristics of the GA, PSO and TLBO approaches for hybridization are analyzed as follows. The GA approach starts its search process using a randomly generated population. The population consists of some individuals with varying fitness values and can provide global searchability, allowing the GA to have more chances to find an optimal solution than local search approaches such as SA and HC. However, since the GA is a population-based search approach and conducts its operators (i.e., selection, crossover and mutation operators) using all individuals in the population, it may have a slow convergence tendency [23]. To mitigate this disadvantage of the GA while maintaining a population-based search, an efficient alternative may be to use only a few individuals in the population, not its entirety [24].

In this paper, the sub-population (50%) with superior fitness values in the initial population is used for the GA loop, because using the sub-population (50%) with superior fitness values improves the chances of finding an optimal solution rather than using the sub-population with inferior fitness values. While the sub-population with superior fitness values may be highly similar and thus, using it may

increase the risk of premature convergence, this similarity and risk can be mitigated using the GA search process because its global searchability can enhance the diversity of individuals in the population.

However, unlike the sub-population with superior fitness values, using the sub-population with inferior fitness values can reduce the chance of finding an optimal solution [23]. This means that if GA starts its search process using the sub-population with inferior fitness values, then the chance of finding an optimal solution is lower than applying its search process using the sub-population with superior fitness values, since GA searches a wide range of values as a way of increasing the diversity of individuals in the population. Therefore, one must actively reduce the diversity of the sub-population with inferior fitness values and therefore, achieve a rapid search process to an optimal solution. Applying PSO to the sub-population with inferior fitness values can be a good alternative because it rapidly improves the fitness values of current individuals using their velocities and positions [23,24].

In this paper, the sub-population (50%) with inferior fitness values in the initial population is used for the PSO loop, because by applying the PSO search process, the fitness values of the individuals in the sub-population may be improved, which makes PSO feasible for rapidly obtaining the optimal solution. In addition, several recent studies [23,24] have shown that further improvement is possible by applying the TLBO search process using elite individuals obtained after applying the GA search process. Since GA lacks an adequate learning ability for the elite individuals during its search process, applying the TLBO search process to some individuals with superior fitness values can provide higher chances to find an optimal solution. In this paper, the sub-offspring (50%) with superior fitness values obtained after GA and PSO loops are used for the TLBO search process.

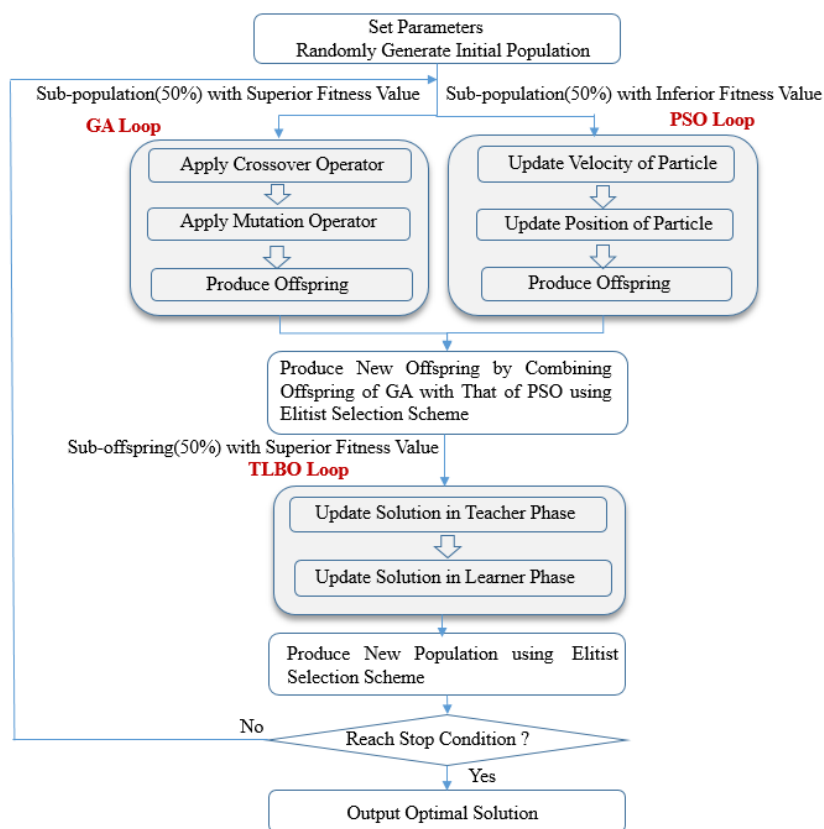


Figure 1. Conceptual flow of the GA-PSO-TLBO approach.

```

input: problem data ( $f(\mathbf{x})$ ,  $n$ ), parameters ( $t$ : maxGen,  $k$ : popSz,  $rand$ : random number)
output: the best solution  $G_{Best}$ 
begin:
   $t \leftarrow 0$ ;
  randomly generate parent population  $P(t) = [\mathbf{x}_k(t)]$ ;
  evaluate  $P(t)$  and keep the best solution  $I_{best}$  in  $P(t)$ ;
  while (not terminating condition) do
    create  $P_1(t)$  using superior solutions (50%) from  $P(t)$ ; //  $P_1(t)=[\mathbf{x}_{k/2}(t)]$ 
    create  $P_2(t)$  using inferior solutions (50%) from  $P(t)$ ; //  $P_2(t)=[\mathbf{x}_{k/2}(t)]$ 
    create  $G(t+1)$  from  $P_1(t)$  by crossover routine; //  $G(t+1)=[\mathbf{x}_{k/2}(t)]$ , GA population // GA loop
    create  $G(t+1)$  from  $P_1(t)$  by mutation routine;
    evaluate  $G(t+1)$  and keep the best solution  $GA_{best}$  in  $G(t+1)$ ;
     $PSO_{best} \leftarrow$  big M; // PSO loop
    for each particle  $\mathbf{x}_k(t)$  in swarm of  $P_2(t)$  do
      update velocity  $\mathbf{v}_k(t)$  by  $\mathbf{v}_k(t+1) = w \mathbf{v}_k + c_1 d_1 (I_{best} - \mathbf{x}_k) + c_2 d_2 (\mathbf{x}_{Best}(t) - \mathbf{x}_k)$ ;
      update position  $\mathbf{x}_k(t+1)$  by  $\mathbf{x}_k(t+1) = \mathbf{x}_k(t) + \mathbf{v}_k(t+1)$ ;
       $S(t+1) \leftarrow \mathbf{x}_k(t+1)$ ; //  $S(t+1)=[\mathbf{x}_{k/2}(t)]$  // PSO population
      if  $PSO_{best} > f(\mathbf{x}_k(t+1))$  then
         $PSO_{best} = \mathbf{x}_k(t+1)$ ;
      end
    end
    create offspring using  $G(t+1)$  and  $S(t+1)$ ; // TLBO loop
    create  $L(t+1)$  using superior solutions (50%) from offspring; //  $L(t+1)=[\mathbf{x}_k(t)]$ , TLBO population
    select teacher value  $X_{best}$  and calculate the mean of the class  $X_{mean}$  in  $L(t+1)$ ;
    for  $k=1$  to popSz/2
       $T_f = round(1 + rand(0,1))$  // Teacher phase
       $\mathbf{x}_k(t+1)^{new} = \mathbf{x}_k(t+1) + rand(X_{best} - T_f X_{mean})$ 
      if  $f(\mathbf{x}_k(t+1)) > f(\mathbf{x}_k(t+1)^{new})$  then
         $\mathbf{x}_k(t+1) = \mathbf{x}_k(t+1)^{new}$ 
      end
      randomly select a learners  $\mathbf{x}_p$  from  $\{1, 2, \dots, popSz\}$ ; // Learner phase
      if  $(\mathbf{x}_k(t+1) > f(\mathbf{x}_p))$  then
         $\mathbf{x}_k(t+1)^{new} = \mathbf{x}_k(t+1) + rand(\mathbf{x}_k(t+1) - \mathbf{x}_p)$ 
      else
         $\mathbf{x}_k(t+1)^{new} = \mathbf{x}_k(t+1) - rand(\mathbf{x}_k(t+1) - \mathbf{x}_p)$ 
      end
      if  $f(\mathbf{x}_k(t+1)) > f(\mathbf{x}_k(t+1)^{new})$  then
         $\mathbf{x}_k(t+1) = \mathbf{x}_k(t+1)^{new}$ 
      end
      select  $X_{best}$  and calculate  $X_{mean}$  in  $L(t+1)$ ;
    end
     $TLBO_{best} = X_{best}$ 
     $G_{best} = \text{argmin}\{I_{best}, GA_{best}, PSO_{best}, TLBO_{best}\}$ 
    reproduce  $P(t+1)$  from  $G(t+1)$ ,  $S(t+1)$ , and  $L(t+1)$  by elitist selection routine;
     $t \leftarrow t+1$ ;
  end
output: the best solution  $G_{Best}$ 
end;

```

Figure 2. Pseudocode for implementing the GA-PSO-TLBO approach.

Secondly, a conceptual flow of the GA-PSO-TLBO approach is shown in Figure 1.

As shown in Figure 1, using a randomly generated initial population, the sub-population (50%) with superior fitness values and the sub-population (50%) with inferior fitness values are applied to

the GA and PSO loops, respectively. The objective of applying GA is to increase the diversity of the population, while that of applying PSO is to enhance the search speed to an optimal solution. By applying GA and PSO together, the exploration and exploitation can be enhanced simultaneously. Using the combined offspring of GA and PSO, the sub-offspring (50%) with superior fitness values are applied to the TLBO loop. The objective of applying TLBO is to increase the chance of finding the optimal solution.

Lastly, for minimization problem, the detailed implementation procedure of the GA-PSO-TLBO approach is described in Figure 2.

4. Case studies

In this section, several well-known engineering optimization problems with various types of design variables are described in detail.

4.1. Type 1 (T-1): Reinforced concrete beam design problem

Introduced by Amir and Hasegawa [3], Shih and Yang [26], Yun [27], and Gandomi et al. [11], T-1 is the problem of designing a reinforced concrete beam. Three types of design variables (continuous, discrete and integer) are used in this problem. It may be assumed that while x_1 can only take 40 numbers of certain predetermined discrete values, x_2 and x_3 can take any continuous values and integer values, respectively. The objective for this problem is to minimize the total cost of concrete and reinforcing steel of the beam. The mathematical formulation is expressed as follows.

$$\min f(\mathbf{x}) = 29.4x_1 + 0.6x_2x_3 \quad (5)$$

$$\text{s. t. } x_1x_3 - (7.735x_1^2 / x_2) - 180 \geq 0 \quad (6)$$

$$(-x_3 / x_2) + 4 \geq 0 \quad (7)$$

where x_1 is discrete, x_2 continuous, and x_3 integer.

4.2. Type 2 (T-2): Gear train design problem

T-2 was introduced by Sandgren [4], and Gandomi et al. [11] and its main objective is to produce a gear ratio as close as possible to 1/6.931. For each gear, the number of teeth must be between 14 and 40, which means that the design variables (x_1, x_2, x_3, x_4) are the numbers of teeth that must be integers. The mathematical formulation is expressed as follows.

$$\min f(\mathbf{x}) = (1/6.931) - (x_1x_2 / x_3x_4)^2 \quad (8)$$

$$\text{s. t. } 12 \leq x_i \leq 60, i = 1,2,3,4 \quad (9)$$

where x_i are all integers.

4.3. Type 3 (T-3): Coil compression spring design problem

T-3 was codified by Wu and Chow [9], and Yun [27]. This problem aims to determine the number of coils (x_1), winding diameter (x_2) and wire diameter (x_3) that result in a minimum mass spring to

carry the given loads while satisfying the design constraints. The design variables are an integer, continuous and discrete, respectively. Especially, discrete design variables (x_3) use predefined discrete dimensions [9]. The mathematical formulation is expressed as follows.

$$\min f(x) = (\pi^2 x_2 x_3^2 (x_1 + 2))/4 \quad (10)$$

$$\text{s. t. } (8C_f F_{max} x_2 / \pi x_3^3) - s \leq 0 \quad (11)$$

$$l_f - l_{max} \leq 0 \quad (12)$$

$$d_{min} - x_3 \leq 0 \quad (13)$$

$$x_2 + x_3 - D_{max} \leq 0 \quad (14)$$

$$3.0 - (x_2/x_3) \leq 0 \quad (15)$$

$$\delta_p - \delta_{pm} \leq 0 \quad (16)$$

$$\delta_p + (F_{max} - F_p)/K + 1.05(x_1 + 2)x_3 - l_f \leq 0 \quad (17)$$

$$\delta_w - (F_{max} - F_p)/K \leq 0 \quad (18)$$

where x_1 is integer, x_2 continuous and x_3 discrete.

The parameters used above are as follows:

$$\delta_p = F_p/K, K = Gx_3^4/8x_1x_2^3, F_{max} = 1,000 \text{ lb}, l_f = F_{max}/K + 1.05(x_1 + 2)x_3,$$

$$S = 289,000 \text{ psi}, l_{max} = 14.0 \text{ inch}, d_{min} = 1.2 \text{ inch}, D_{max} = 3.0 \text{ inch}, F_p = 300.0 \text{ lb},$$

$$\delta_{pm} = 6.0 \text{ inch}, \delta_w = 1.25 \text{ inch}, G = 11.5 \times 10^6 \text{ psi}, C_f = (4(x_2/x_3) - 1)/(4(x_2/x_3) - 4) - 0.165x_3/x_2.$$

4.4. Type 4 (T-4): Corrugated bulkhead design problem

Initially studied by Kvalie [28], T-4 aims to minimize the weight of corrugated bulkheads for a tanker. The design variables used in T-4 are width (x_1), depth (x_2), length (x_3), and plate thickness (x_4). The minimum-weight requires the solution of the following optimization problem [11]. The mathematical formulation is expressed as follows.

$$\min f(x) = (5.885x_4(x_1 + x_3))/(x_1 + \sqrt{(x_3^2 - x_2^2)}) \quad (19)$$

$$\text{s. t. } x_4x_2(0.4x_1 + x_3/6) - 8.94(x_1 + \sqrt{(x_3^2 - x_2^2)}) \geq 0 \quad (20)$$

$$x_4x_2^2(0.2x_1 + (x_3/12)) - 2.2(8.94(x_1 + \sqrt{(x_3^2 - x_2^2)}))^{4/3} \geq 0 \quad (21)$$

$$x_4x_2^2(0.2x_1 + (x_3/12)) - 2.2(8.94(x_1 + \sqrt{(x_3^2 - x_2^2)}))^{4/3} \geq 0 \quad (22)$$

$$x_4 - 0.0156x_1 - 0.15 \geq 0 \quad (23)$$

$$x_4 - 0.0156x_3 - 0.15 \geq 0 \quad (24)$$

$$x_4 - 1.05 \geq 0 \quad (25)$$

$$x_3 - x_2 \geq 0 \quad (26)$$

where $x_1 \geq 0$, $x_2, x_3 \leq 100$, $0 \leq x_4 \leq 5$, and x_i are all continuous

4.5. Type 5 (T-5): Pressure vessel design problem

T-5 was studied by Sandgran [4], Fu et al. [5], Wu and Chow [9], and Gandomi et al. [11]. The design variables are the dimensions required for the specifications of vessel. The design variables x_1 and x_2 are discrete values and have integral multipliers of 0.0625 inch, respectively. The design variables x_3 and x_4 are continuous values, and the side constraints are $40 \leq x_3 \leq 80$ inches and $20 \leq x_4 \leq 60$ inches, respectively. The mathematical formulation is expressed as follows.

$$\min f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (27)$$

$$\text{s. t. } x_1 - 0.0193x_3 \geq 0 \quad (28)$$

$$x_2 - 0.00954x_3 \geq 0 \quad (29)$$

$$\pi x_3^2x_4 + (4/3)\pi x_3^3 - (750 \times 1782) \geq 0 \quad (30)$$

$$-x_4 + 240 \geq 0 \quad (31)$$

$$x_1 - 1.1 \geq 0 \quad (32)$$

$$x_2 - 0.6 \geq 0 \quad (33)$$

where x_1 and x_2 are discrete, and x_3 and x_4 continuous.

4.6. Type 6 (T-6): Speed reducer design problem

T-6 is a benchmark engineering optimization problem proposed by Ray and Saini [29], and Gandomi and Yang [30]. Seven design variables are used in total: face width (x_1), module of teeth (x_2), number of teeth on pinion (x_3), length of shaft 1 between bearings (x_4), length of shaft 2 between bearings (x_5), diameter of shaft 1 (x_6), and diameter of shaft 2 (x_7). The objective is to minimize the total weight of the speed reducer. The constraints involve limitations on the bending stress of the gear teeth, surface stress, transverse deflections of shafts 1 and 2 due to transmitted force, and stresses in shafts 1 and 2. The mathematical formulation is expressed as follows.

$$\min f(\mathbf{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \quad (34)$$

$$\text{s. t. } 27x_1^{-1}x_2^{-2}x_3^{-1} \leq 1 \quad (35)$$

$$397.5x_1^{-1}x_2^{-2}x_3^{-2} \leq 1 \quad (36)$$

$$1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \leq 1 \quad (37)$$

$$1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \leq 1 \quad (38)$$

$$((745x_4/x_2x_3)^2 + (1.69)10^6)^{0.5}/0.1x_6^3 \leq 1,100 \quad (39)$$

$$((745x_5/x_2x_3)^2 + (157.5)10^6)^{0.5}/0.1x_7^3 \leq 850 \quad (40)$$

$$x_2x_3 \leq 40 \quad (41)$$

$$5 \leq x_1x_2 \leq 12 \quad (42)$$

$$(1.5x_6 + 1.9)x_4^{-1} \leq 1 \quad (43)$$

$$(1.1x_7 + 1.9)x_5^{-1} \leq 1 \quad (44)$$

where $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4, x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, $5.0 \leq x_7 \leq 5.5$ and x_i are all continuous.

Using the six aforementioned engineering optimization problems, the performance of the GA-PSO-TLBO approach is compared with that of several single and hybrid metaheuristic approaches in Table 1.

Table 1. Single and hybrid metaheuristic approaches for comparison.

| Type | Approach | Reference |
|-------------------------------|----------|--|
| Single metaheuristic approach | GA | Gen and Cheng [31] |
| | PSO | Kennedy and Eberhart [32], Yu and Gen [33] |
| | TLBO | Rao [34] |
| Hybrid metaheuristic approach | GA-PSO | Kao and Zahara [23] |
| | GA-TLBO | Güçyetmez and Çam [25] |

Table 2. Parameter setting values and operators.

| | |
|---|------------------------------------|
| For GA search process | |
| Population size | 20 |
| Crossover rate | 0.5 |
| Mutation rate | 0.3 |
| Selection Operator | Elitist selection scheme [35] |
| Crossover operator | Non-uniform arithmetic scheme [36] |
| Mutation operator | Random mutation scheme [18] |
| For PSO search process | |
| Number of iterations | 100 |
| Number of individuals (= population size) | 20 |
| For TLBO search process | |
| Number of iterations | 100 |
| Population size | 20 |

All approaches shown in Table 1, including the GA-PSO-TLBO approach, were programmed using MATLAB version 2014b and were simulated on an IBM-compatible PC having a 1.3 GHz processor with 4 GB of RAM. The parameter setting values and operators to implement each approach are summarized in Table 2.

All setting values were obtained after fine-tuning process of each approach. Total number of iterations is 1000. All results are obtained after 10 independent runs to eliminate the randomness in the run of each approach. Three measures of performance are used to compare the performances of all approaches, as shown in Table 3.

Table 3. Measures of performance.

| Measure | Description |
|---------|---|
| BS | Best solution in all runs |
| AS | Average solution through all runs |
| CPU | Averaged running time through all runs (unit: sec.) |

Tables 4 and 5 show the comparison results among conventional single metaheuristic approaches (GA, PSO, and TLBO), conventional hybrid metaheuristic approaches (GA-PSO and GA-TLBO) and the GA-PSO-TLBO approach. It can be observed that the performances of GA, PSO, TLBO, GA-PSO and GA-TLBO approaches are worse than that of the GA-PSO-TLBO approach if plus (+) values appeared in Gaps 1 and 2, but the former are better than the latter if minus (−) values appeared in Gaps 1 and 2.

In T-1 of Table 4, the differences between the GA-PSO-TLBO approach and the GA, PSO and TLBO approaches are 0.453, 1.810 and 0.002 %, respectively, in terms of the BS. This means that the GA-PSO-TLBO approach has slightly better performance than the TLBO approach, but has significantly better performance than the GA and PSO approaches. Similar situations are also seen in terms of the AS—the differences among the GA, PSO and TLBO approaches are 2.035, 3.448 and 0.022%, respectively, when compared with the GA-PSO-TLBO approach, indicating that the performance of the GA-PSO-TLBO approach is slightly better than that of the TLBO approach, but significantly better than those of the GA and PSO approaches. However, in terms of CPU, the GA-PSO-TLBO approach is the slowest, and the TLBO approach is the quickest.

In T-2, T-3, T-5 and T-6 of Table 4, the performance metrics of the GA-PSO-TLBO approach are considerably superior to those of the other approaches in terms of the BS and AS, but its search speed is the slowest in terms of the CPU. In T-4 of Table 4, the GA-PSO-TLBO approach shows better performances in terms of the BS and AS than the GA and PSO approaches, but the former does not show any merit in terms of the BS when compared with the TLBO approach.

In T-1 of Table 5, the differences between the GA-PSO-TLBO approach and the GA-PSO and GA-TLBO approaches are 0.205 and 0.003 %, respectively, in terms of the BS, which means that the GA-PSO-TLBO approach has slightly better performance than the GA-TLBO approach, but significantly better performance than the GA-PSO approach. Similar situations are also seen in terms of the AS—the differences of the GA-PSO and GA-TLBO approaches when compared with the GA-PSO-TLBO approach are 3.748, and 0.356%, respectively, which indicates that the performance of the GA-PSO-TLBO approach is considerably better than those of the GA-PSO and GA-TLBO approaches. However, in terms of the CPU, the GA-PSO-TLBO approach has the slowest performance, and the GA-TLBO approach has the quickest performance.

In T-2, T-3, T-5 and T-6, but not in T-4, of Table 5, the GA-PSO-TLBO approach outperforms those of the other approaches in terms of the BS and AS, but the search speed of the former is the slowest in terms of CPU.

Table 4. Comparison results between conventional single metaheuristic approaches (GA, PSO and TLBO) and the proposed GA-PSO-TLBO approach.

| Case Study | GA | | | | | PSO | | | | |
|------------|------------------------|-----------------------|-------------|-----------|------------|-----------------------|-----------------------|-------------|-----------|-----------|
| | BS | AS | CPU | Gap 1 (%) | Gap 2 (%) | BS | AS | CPU | Gap 1 (%) | Gap 2 (%) |
| T-1 | 366.508 | 372.280 | 2.30 | 0.453 | 2.035 | 371.579 | 377.434 | 8.00 | 1.843 | 3.448 |
| T-2 | 9.92×10^{-10} | 6.18×10^{-7} | 2.30 | 36634.929 | 463126.866 | 1.26×10^{-9} | 1.69×10^{-8} | 1.00 | 46676.935 | 12595.401 |
| T-3 | 2.452 | 3.581 | 1.50 | 17.768 | 70.299 | 2.512 | 2.883 | 6.00 | 20.634 | 37.098 |
| T-4 | 7.123 | 7.305 | 0.40 | 4.090 | 6.757 | 7.175 | 7.651 | 0.80 | 4.848 | 11.807 |
| T-5 | 7264.114 | 7351.335 | 1.60 | 0.918 | 2.021 | 7309.212 | 8181.397 | 1.60 | 1.545 | 13.540 |
| T-6 | 2925.490 | 2951.542 | 0.50 | 1.082 | 1.982 | 3070.831 | 4299.165 | 1.50 | 6.104 | 48.546 |

| Case Study | TLBO | | | | | GA-PSO-TLBO | | |
|------------|------------------------|------------------------|-------------|-----------|-----------|--|--|-------|
| | BS | AS | CPU | Gap 1 (%) | Gap 2 (%) | BS | AS | CPU |
| T-1 | 364.860 | 364.935 | 1.60 | 0.002 | 0.022 | 364.854 | 364.854 | 3.10 |
| T-2 | 2.31×10^{-11} | 7.53×10^{-10} | 1.30 | 754.475 | 464.625 | 2.70×10^{-12} | 1.33×10^{-10} | 2.40 |
| T-3 | 2.275 | 2.672 | 43.50 | 9.262 | 27.085 | 2.082 | 2.103 | 47.90 |
| T-4 | 6.843 | 6.843 | 0.89 | 0.000 | 0.000 | 6.843 | 6.843 | 2.30 |
| T-5 | 7225.661 | 7337.083 | 1.70 | 0.384 | 1.823 | 7198.010 | 7205.729 | 2.90 |
| T-6 | 2894.991 | 2912.423 | 1.30 | 0.028 | 0.631 | 2894.172 | 2894.172 | 1.80 |

* Best results in each row are bold.

* Gap 1(%): Difference when compared the performances of the GA, PSO and TLBO approaches with that of the GA-PSO-TLBO approach in terms of the BS.

* Gap 2(%): Difference when compared the performances of the GA, PSO and TLBO approaches with that of the GA-PSO-TLBO approach in terms of the AS

Table 5. Comparison results between conventional hybrid metaheuristic approaches (GA-PSO and GA-TLBO) and the proposed GA-PSO-TLBO approach.

| Case Study | GA-PSO | | | | | GA-TLBO | | | | | GA-PSO-TLBO | | | | |
|------------|------------------------|-----------------------|--------------|-----------|-----------|------------------------|-----------------------|-------------|-----------|-----------|--|--|-------|--|--|
| | BS | AS | CPU | Gap 1 (%) | Gap 2 (%) | BS | AS | CPU | Gap 1 (%) | Gap 2 (%) | BS | AS | CPU | | |
| T-1 | 365.603 | 378.528 | 3.80 | 0.205 | 3.748 | 364.866 | 366.154 | 2.60 | 0.003 | 0.356 | 364.854 | 364.854 | 3.10 | | |
| T-2 | 2.31×10^{-11} | 2.24×10^{-8} | 1.10 | 754.475 | 16713.483 | 2.31×10^{-11} | 1.74×10^{-8} | 1.10 | 754.475 | 12929.190 | 2.70×10^{-12} | 1.33×10^{-10} | 2.40 | | |
| T-3 | 2.087 | 2.267 | 40.80 | 0.227 | 7.824 | 2.127 | 2.649 | 87.20 | 2.132 | 26.009 | 2.082 | 2.103 | 47.90 | | |
| T-4 | 6.934 | 7.066 | 1.40 | 1.330 | 3.261 | 6.843 | 6.845 | 2.30 | 0.000 | 0.024 | 6.843 | 6.843 | 2.30 | | |
| T-5 | 7291.471 | 7435.440 | 1.10 | 1.298 | 3.188 | 7201.860 | 7527.609 | 2.10 | 0.053 | 4.467 | 7198.010 | 7205.729 | 2.90 | | |
| T-6 | 2912.937 | 2980.467 | 1.20 | 0.648 | 2.982 | 2894.895 | 3013.943 | 1.20 | 0.025 | 4.138 | 2894.172 | 2894.172 | 1.80 | | |

* Best results in each row are bold.

* Gap 1(%): Difference when compared the performances of the GA-PSO and GA-TLBO approaches with that of the GA-PSO-TLBO approach in terms of the BS.

* Gap 2(%): Difference when compared the performances of the GA-PSO and GA-TLBO approaches with that of the GA-PSO-TLBO approach in terms of the AS.

For a more detailed comparison, Figures 3 and 4 show the convergence behaviors of all the approaches when the number of iterations is increased to 100.

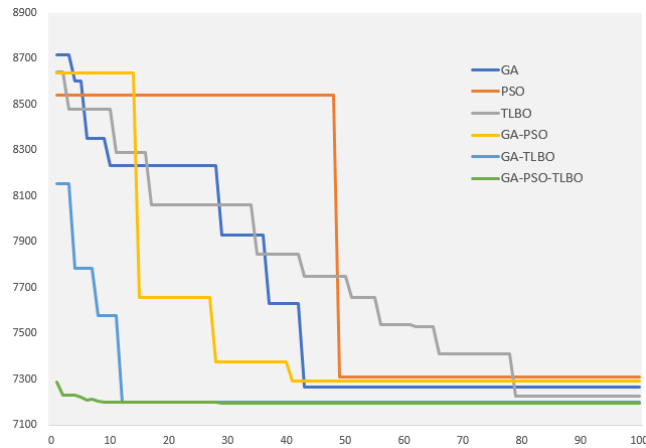


Figure 3. Convergence behaviors of each approach for T-5.

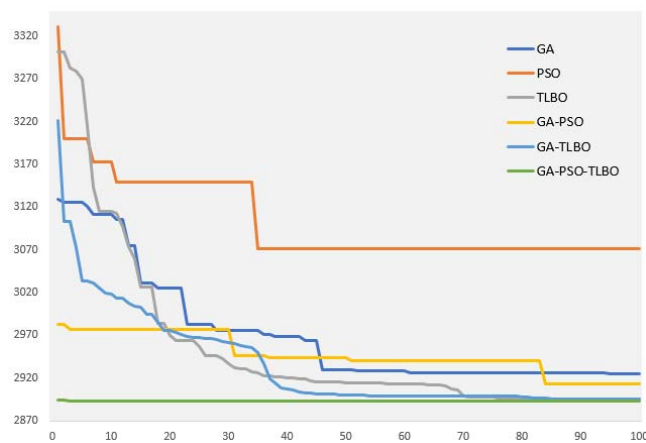


Figure 4. Convergence behaviors of each approach for T-6.

It is evident from Figure 3 that the GA-PSO-TLBO approach rapidly enhances the optimization in the early stage, while all other approaches (GA, TLBO, GA-PSO and GA-TLBO), except the PSO approach, show various convergence behaviors in their early stages, but inferior overall performance to that of the GA-PSO-TLBO approach in the later stages. Similar convergence behaviors are also shown in Figure 4, where all the competing approaches, except the PSO approach, show various convergence behaviors in their early stages, while the GA-PSO-TLBO approach shows rapid convergence behavior compared to the others, though the former does not show any different convergence behavior than the others in its early stage.

To compare the stability of each approach, the distributions of the best solutions obtained after 10 independent runs for T-5 and T-6 are shown in Figures 5 and 6.

These figures show that the GA-PSO-TLBO approach can provide a more compact distribution with a lower average value than the GA, PSO, TLBO, GA-PSO and GA-TLBO approaches, which indicates its reliability and satisfactory optimization ability.

Through the computation results of Tables 4 and 5 and Figures 3–6, the following conclusions can be reached.

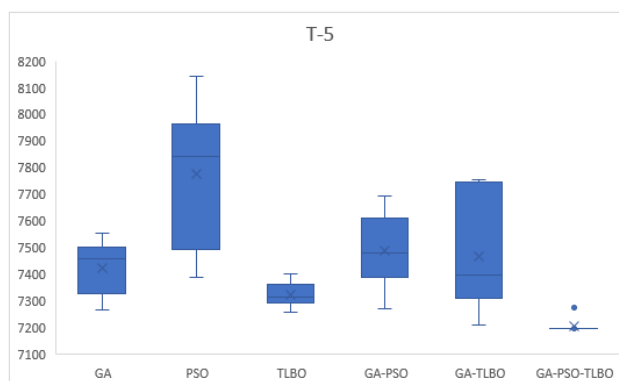


Figure 5. Distributions of the best solutions at each approach for T-5.

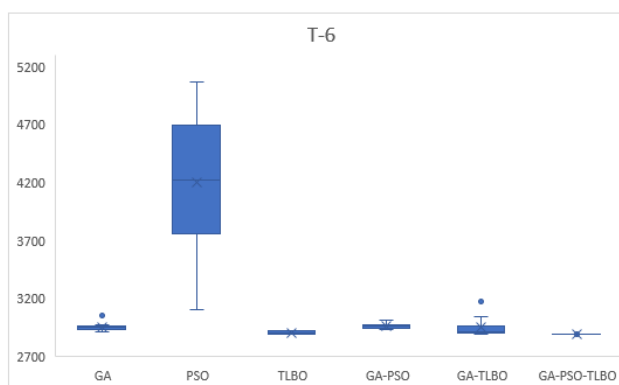


Figure 6. Distributions of the best solutions at each approach for T-6.

- Among all single metaheuristic approaches using T-1, T-2, T-4, T-5, and T-6, except T-3 in Table 4, the TLBO approach has shown better performance in terms of BS and AS than the GA and PSO approaches, indicating that the search scheme used in the TLBO approach is more efficient than those used in the GA and PSO approaches.
- In the comparison among all hybrid metaheuristic approaches in Table 5, it has been observed that the performance metrics of the GA-TLBO approach are superior to those of the GA-PSO approach in terms of BS in T-4, T-5, and T-6, but not in terms of BS and AS in T-3. This implies that the search scheme used in the GA-TLBO approach is usually more efficient in finding better solutions than that used in the GA-PSO approach.
- In the comparison between single metaheuristic approaches (GA, PSO, and TLBO) and the hybrid ones (GA-PSO and GA-TLBO) of Tables 4 and 5, the latter do not show any merits compared with the former, since some computation results (in terms of the BS and AS for T-1, and AS for T-4, T-5, and T-6) of the TLBO approach have shown better performances than the GA-PSO and GA-TLBO approaches. This means that the hybridization schemes used in the GA-PSO and GA-TLBO approaches do not show sufficient advantages in searching the best

solutions in T-1 to T-6 when compared with the GA, PSO, and TLBO approaches, though the former have more search schemes than the latter.

- In the comparison among all approaches, the GA-PSO-TLBO approach outperforms both single metaheuristic approaches (GA, PSO, and TLBO) and hybrid ones (GA-PSO and GA-TLBO) in terms of the BS and AS, except in the case of T-4. This analysis result shows that the success of a hybrid metaheuristic approach, like the GA-PSO-TLBO approach, depends highly on how single metaheuristic approaches are combined.

4.7. Additional analysis of the GA-PSO-TLBO approach

In the GA-PSO-TLBO approach, the sub-population (50%) with superior fitness values is used for the GA loop, while the other sub-population with inferior fitness values is used for the PSO loop. However, some conventional studies [13,18,37–39] have used the whole population when they combined several metaheuristic approaches. Yun et al. [18] proposed a GA-CS approach, where the whole population generated at the initial stage was first used for GA loop and then the CS loop was applied using all offspring obtained after the GA loop. Similar studies using the whole population in each metaheuristic approach have also been performed by Castelli and Vanneschi [13] and Zhai et al. [38] who combined the GA approach with the VNS approach and Marinakis and Marinaki [37] who combined the GA approach with the PSO approach. Recently, Gen, et al. [39] proposed hybridized metaheuristics by TLBO with GA and PSO and applied to supply chain network model as one of case studies.

Therefore, it would be worthwhile to compare the performance between the studies using the whole population and the studies using a part of the population in hybrid metaheuristic approaches. For achieving this purpose, the GA-PSO-TLBO approach is divided into two parts, as follows:

- GA-PSO-TLBO1: the whole population (100%) randomly generated in the initial stage is used for the GA and PSO loops. Also, the whole offspring (100%) obtained after the GA and PSO loops is used for the TLBO loop.
- GA-PSO-TLBO2: the sub-population (50%) with superior fitness values is used for the GA loop, while the other sub-population (50%) with inferior fitness values is used for the PSO loop. However, the whole offspring (100%) obtained after the GA and PSO loops is used for the TLBO loop.

Table 6. Comparison results between the GA-PSO-TLBO1 and GA-PSO-TLBO approaches.

| Case Study | GA-PSO-TLBO1 | | | | GA-PSO-TLBO | | |
|---------------|-----------------|-----------------|------|-----------|-----------------|-----------------|-------------|
| | BS | AS | CPU | Gap 3 (%) | BS | AS | CPU |
| T-4 | 6.843 | 6.843 | 5.30 | 130.43 | 6.843 | 6.843 | 2.30 |
| T-6 | 2894.172 | 2894.172 | 4.70 | 161.11 | 2894.172 | 2894.172 | 1.80 |

* Best results in each row are bold.

* Gap 3(%): Difference when compared the performances of the GA-PSO-TLBO1 approach with that of the GA-PSO-TLBO approach in terms of the CPU.

The main difference between the GA-PSO-TLBO and GA-PSO-TLBO1 is that the former uses a part of the population in the GA, PSO and TLBO loops, while the latter uses the whole population for those loops. Also, the main difference between the GA-PSO-TLBO and GA-PSO-TLBO2 is that the

former uses a part of the whole population in the TLBO loop, while the latter uses the whole population as it is for the same loop. Tables 6 and 7 show the computation results using T-4 and T-6 for the GA-PSO-TLBO, GA-PSO-TLBO1, and GA-PSO-TLBO2 approaches.

Table 7. Comparison results between the GA-PSO-TLBO2 and GA-PSO-TLBO approaches.

| Case Study | GA-PSO-TLBO2 | | | | GA-PSO-TLBO | | |
|---------------|-----------------|-----------------|------|-----------|-----------------|-----------------|-------------|
| | BS | AS | CPU | Gap 3 (%) | BS | AS | CPU |
| T-4 | 6.843 | 6.843 | 3.50 | 52.17 | 6.843 | 6.843 | 2.30 |
| T-6 | 2894.172 | 2894.172 | 2.40 | 33.33 | 2894.172 | 2894.172 | 1.80 |

* Best results in each row are bold.

* Gap 3(%): Difference when compared the performances of the GA-PSO-TLBO2 approach with that of the GA-PSO-TLBO approach in terms of the CPU.

In Table 6, it is observed that the performance of the GA-PSO-TLBO approach is equal to that of the GA-PSO-TLBO1 approach in terms of the BS and AS, but in terms of the CPU, the former shows significantly quicker search speed than the latter, indicated by the gaps between them being 130.43 (%) for T-4 and 161.11 (%) for T-6. This implies that using sub-population (50%) and sub-offspring (50%) for the GA, PSO and TLBO loops is more efficient in terms of the CPU than using the whole population and offspring, though they have the same results in terms of the BS and AS.

In Table 7, both the GA-PSO-TLBO and GA-PSO-TLBO2 approaches have the same results in terms of the BS and AS, but the former shows considerably better performance in terms of the CPU than the latter, with the differences between them being 52.17 (%) for T-4 and 33.33 (%) for T-6. This means that applying the sub-offspring (50%) obtained after GA and PSO loops to TLBO loop is more efficient than applying the whole offspring obtained after GA and PSO loops to the TLBO loop.

Through the computation results of Tables 6 and 7, it can be concluded that when combining GA, PSO and TLBO for the GA-PSO-TLBO approach, using sub-population (50%) of the whole population obtained in the initial stage for GA and PSO loops as well as using sub-offspring (50%) of the whole offspring obtained after GA and PSO loops for the TLBO loop are more efficient in reducing search speed than using the whole population and offspring for all three loops.

5. Conclusions

Over the last three decades, many researchers have developed various approaches to efficiently solve engineering optimization problems. Most conventional studies typically either convert the type of design variables or introduce a penalty function to the original design variables. However, this either results in deterioration of the quality of the obtained solution, or occurrence of premature convergence through the conversion or introduction process. To overcome this disadvantage, various metaheuristic approaches have been developed. The primary advantage of applying metaheuristic approaches to engineering optimization problems is that the original design variable types can be used as without any conversion or introduction process.

In this study, we developed a new hybrid metaheuristic approach called the GA-PSO-TLBO approach. This approach combines the merits of three single metaheuristic approaches, that is, GA approach with global searchability, PSO approach with local searchability, and TLBO approach with

learning ability. To improve the time-efficiency of the GA-PSO-TLBO approach, the sub-population (50%) with superior fitness values, sub-population (50%) with inferior fitness values of the whole population obtained in the initial stage, and sub-offspring (50%) with superior fitness values in the whole offspring obtained after the GA and PSO loops, are used for the GA, PSO, and TLBO loops, respectively.

The performance of the GA-PSO-TLBO approach was compared with those of three single metaheuristic approaches (GA, PSO, and TLBO), respectively and two hybrid metaheuristic approaches (GA-PSO, and GA-TLBO) using six well-known engineering optimization problems. The experimental results have proved and demonstrated that the GA-PSO-TLBO approach outperforms the GA, PSO, TLBO, GA-PSO and GA-TLBO approaches in terms of the BS and AS, which implies that the success of a hybrid metaheuristic approach like the GA-PSO-TLBO approach depends highly on how the merits of each constituent metaheuristic approach are combined.

For additional analysis of the GA-PSO-TLBO approach, two competing approaches (GA-PSO-TLBO1 and GA-PSO-TLBO2) have been suggested, and the experimental results have proved that the performance of the GA-PSO-TLBO approach is superior to those of the GA-PSO-TLBO1 and GA-PSO-TLBO2 approaches in terms of the CPU, which implies that when combining GA, PSO, and TLBO for the GA-PSO-TLBO approach, using a sub-population (50%) of the whole population obtained in the initial stage for the GA and PSO loops as well as using a sub-offspring (50%) of the whole offspring obtained after the GA and PSO loops for the TLBO loop, increase the efficiency and reduce the search time than using the whole population and offspring for the GA, PSO and TLBO loops.

However, although the GA-PSO-TLBO approach combines the merits of the GA, PSO, and TLBO approaches, its search speed is slower than those of three single metaheuristic approaches (GA, PSO, and TLBO) and two hybrid metaheuristic approaches (GA-PSO, and GA-TLBO). Therefore, a method to increase the search speed of the GA-PSO-TLBO approach must be found and more comparisons between the GA-PSO-TLBO approach and other hybrid metaheuristic approaches such as GA-SA-TLBO or GA-TS (Tabu search)-TLBO approaches must be conducted; these are potential avenues of future research in this domain.

Acknowledgments

This work was supported by research fund from Chosun University (2022).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. M. Gen, Y. Yun, Soft computing approach for reliability optimization: State-of-the-art survey, *Rel. Eng. and Sys. Saf.*, **91** (2006), 1008–1026. <https://doi.org/10.1016/j.res.2005.11.053>
2. G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, *Trans. ASME*, **129** (2007), 370–380. <https://doi.org/10.1115/1.2429697>

3. H. M. Amir, T. Hasegawa, Nonlinear mixed-discrete structural optimization, *J. Struct. Eng.*, **115** (1989), 626–646. [https://doi.org/10.1061/\(ASCE\)0733-9445\(1989\)115:3\(626\)](https://doi.org/10.1061/(ASCE)0733-9445(1989)115:3(626))
4. E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, *ASME J. Mech. Des.*, **112** (1990), 223–229. <https://doi.org/10.1115/1.2912596>
5. J. F. Fu, R. G. Fenton, W. L. Cleghorn, A mixed integer-discrete-continuous programming method and its applications to engineering design optimization, *Eng. Opt.*, **17** (1991), 263–280. <https://doi.org/10.1080/03052159108941075>
6. W. Kuo, V. R. Prasad, F. Tillman, C. L. Hwang, *Optimal reliability design: Fundamentals and applications*, Cambridge University Press, 2001.
7. N. G. Yarushkina, Genetic algorithms for engineering optimization: Theory and practice, in *Proceedings of the 2002 IEEE International Conference on Artificial Intelligence System (ICAIS'02)*, (2002), 357–362. <https://doi.org/10.1109/ICAIS.2002.1048127>
8. C. -Y. Lin, P. Hajela, Genetic algorithms in optimization problems with discrete and integer design variables, *Eng. Opt.*, **19** (1992), 309–327. <http://doi.org/10.1080/03052159208941234>
9. S. J. Wu, P. T. Chow, Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization, *Eng. Opt.*, **24** (1995), 137–159. <http://doi.org/10.1080/03052159508941187>
10. T. Yokota, T. Taguchi, M. Gen, A solution method for optimal weight design problem of the gear using genetic algorithms, *Comp. Ind. Eng.*, **35** (1998), 523–526. [http://doi.org/10.1016/S0360-8352\(98\)00149-1](http://doi.org/10.1016/S0360-8352(98)00149-1)
11. A. H. Gandomi, X-S. Yang, A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comp.*, **29** (2013), 17–35. <http://doi.org/10.1007/s00366-011-0241-y>
12. E. S. Maputi, R. Arora, Design optimization of a three-stage transmission using advanced optimization techniques, *Int. J. Simul. Multidisci. Des. Opt.*, **10** (2019). <http://doi.org/10.1051/smdo/2019009>
13. M. Castelli, L. Vanneschi, Genetic algorithm with variable neighborhood search for the optimal allocation of goods in shop shelves, *Oper. Res. Lett.*, **42** (2014), 355–360. <http://doi.org/10.1016/j.orl.2014.06.002>
14. S. Babaie-Kafaki, R. Ghanbari, N. Mahdavi-Amiri, Hybridizations of genetic algorithms and neighborhood search metaheuristics for fuzzy bus terminal location problems, *App. Soft Comp.*, **46** (2016), 220–229. <http://doi.org/10.1016/j.asoc.2016.03.005>
15. O. Dib, M-A. Manier, L. Moalic, A. Caminada. Combining VNS with genetic algorithm to solve the one-to-one routing issue in road networks, *Comp. Oper. Res.*, **78** (2017), 420–430. <http://doi.org/10.1016/j.cor.2015.11.010>
16. O. Dib, A. Moalic, M-A. Manier, A. Caminada. An advanced GA–VNS combination for multicriteria route planning in public transit networks, *Exp. Sys. with Appl.*, **72** (2017), 67–82. <http://doi.org/10.1016/j.eswa.2016.12.009>
17. M. Gen, L. Lin, Y. Yun, H. Inoue, Recent advances in hybrid priority-based genetic algorithms for logistics and SCM network design, *Comp. Ind. Eng.*, **115** (2018), 394–412. <http://doi.org/10.1016/j.cie.2018.08.025>
18. Y. Yun, A. Chuluunsukh, M. Gen, Sustainable closed-loop supply chain design problem: A hybrid genetic algorithm approach, *Mathematics*, **8** (2020), 84. <http://doi.org/10.3390/math8010084>

19. I. Sbai, S. Krichen, O. Limam, Two meta-heuristics for solving the capacitated vehicle routing problem: the case of the Tunisian Post Office, *Oper. Res.*, **20** (2020), 2085–2108. <http://doi.org/10.1007/s12351-019-00543-8>
20. J. Wu, M. Fan, Y. Liu, Y. Zhou, N. Yang, M. Yin, A hybrid ant colony algorithm for the winner determination problem, *Math. Biosci. Eng.*, **19** (2022), 3202–3222. <http://doi:10.3934/mbe.2022148>
21. Y. Yun, C. U. Moon, Comparison of adaptive genetic algorithm for engineering optimization problems, *Int. J. Ind. Eng.*, **10** (2003), 584–590.
22. K. Nitisiri, H. Ohwada, M. Gen, Hybrid genetic algorithm with auto-tuning parameters and K-mean clustering strategy for multiple optimization, *J. Soc. Plant Eng. Japan*, **31** (2019), 58–67.
23. Y-T. Kao, E. Zahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, *Appl. Soft Comp.*, **8** (2008), 849–857. <http://doi.org/10.1016/j.asoc.2007.07.002>
24. X. Huang, Z. Guan, L. Yang, An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem, *Adv. Mech. Eng.*, **10** (2018), 1–14. <http://doi.org/10.1177/1687814018801442>
25. M. Güçyetmez, E. Çam, A new hybrid algorithm with genetic-teaching learning optimization (G-TLBO) technique for optimizing of power flow in wind-thermal power systems, *Elect. Eng.*, **98** (2016), 145–157. <http://doi.org/10.1007/s00202-015-0357-y>
26. C. J. Shih, Y. C. Yang, Generalized Hopfield network based structural optimization using sequential unconstrained minimization technique with additional penalty strategy, *Adv. Eng. Sof.*, **33** (2002), 721–729. [http://doi.org/10.1016/S0965-9978\(02\)00060-1](http://doi.org/10.1016/S0965-9978(02)00060-1)
27. Y. Yun, *Study on adaptive hybrid genetic algorithm and its applications to engineering design problems*, Ph.D. Thesis, Waseda University, Japan, 2005.
28. D. Kvalie, *Optimization of plane elastic grillages*, PhD Thesis, Norges Teknisk Naturvitenskapelige Universitet, Norway, 1967.
29. T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, *Eng. Opt.*, **33** (2007), 735–748. <http://doi.org/10.1080/03052150108940941>
30. A. H. Gandomi, X. S. Yang, *Benchmark problems in structural optimization, Chapter 12 in Comp. Opt., Meth. and Alg.*, (eds. S. Koziel, X-S. Yang) Springer-Verlag, Berlin, (2011), 267–291. http://doi.org/10.1007/978-3-642-20859-1_12
31. M. Gen, R. Cheng, *Genetic algorithms and engineering optimization*, John Wiley & Sons, New York, NY, USA, 2000.
32. J. Kennedy, R. C. Eberhart, Particle swarm optimization, in *Proceedings on IEEE International Conference on Neural Networks*, (1995), 1942–1948. <http://doi:10.1109/ICNN.1995.488968>
33. X. Yu, M. Gen, *Introduction to evolutionary algorithms*, Springer, London, UK, 2010.
34. R. V. Rao, *Teaching learning based optimization algorithm and its engineering applications*, Springer, Switzerland, 2016. http://doi.org/10.1007/978-3-319-22732-0_2
35. M. Gen, R. Cheng, *Genetic algorithms and engineering design*, John Wiley and Sons, New York, 1997.
36. Z. Michalewicz, *Genetic algorithms + data structures = evolution program*, Spring-Verlag, 1994.
37. Y. Marinakis, M. Marinaki, A hybrid genetic - Particle swarm optimization algorithm for the vehicle routing problem, *Exp. Syst. Appl.*, **37** (2010), 1446–1455. <http://doi.org/10.1016/j.eswa.2009.06.085>

38. H. Zhai, Y. K. Liu, K. Yang, Modeling two-stage UHL problem with uncertain demands, *Appl. Math. Model.*, **40** (2016), 3029–2048. <http://doi.org/10.1016/j.apm.2015.09.086>
39. M. Gen, A. Chuluunsukh, Y. Yun, Hybridizing teaching-learning based optimization with GA and PSO: Case study of supply chain network model, in *the 2021 International Conference on Computational Science and Computational Intelligence (CSCI 2021)*, Las Vegas, USA, (2021). <http://doi:10.1109/CSCI54926.2021.00146>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)