International Journal of Innovative Computing, Information and Control
Volume **8**, Number **6**, June **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3997–4014

EFFECTIVE CONSTRAINTS BASED EVOLUTIONARY ALGORITHM FOR CONSTRAINED OPTIMIZATION PROBLEMS

Sanghoun Oh[1], Chang Wook Ahn[2] and Moongu Jeon[1,*]

[1]School of Information and Communications
Gwangju Institute of Science and Technology
123 Cheomdan-gwagiro (Oryong-dong), Buk-gu, Gwangju 500-712, Korea
oosshoun@gist.ac.kr; *Corresponding author: mgjeon@gist.ac.kr

[2]School of Information and Communication Engineering
Sungkyunkwan University
300 Cheonchoen-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
cwan@skku.edu

Received February 2011; revised June 2011

Abstract. *Evolutionary algorithms (EAs) have an enviable success record in solving many constrained optimization problems (COPs) in a variety of areas. However, those problems are afflicted by the highly constrained feasibility – isolated and small feasible regions, since EAs should thoroughly test feasibility for all constraints at every generation. To systematically deal with its limitation, this paper presents a new approach for effectively handling the constraints of COPs. The major idea is to extract an actual subset of meaningful constraints, termed "effective constraints", from the current population. This discovered set plays a key role in satisfying the feasibility within the certain tolerance specified by the statistics on feasible solutions and several prefixed criteria on feasibility. Thanks to the proposed effective constraints, it is able to evolve the population toward the legitimate region of the search space without assessing all the constraints; thus, the better feasible space is yielded than that of originals. The proposed constraint-handling technique is combined with a widely-used EA, stochastic ranking evolutionary strategy, for achieving the optimal solutions of COPs. The proposed algorithm is compared with several well-known references on three real-world engineering optimization problems. Computational studies verify that the proposed algorithm achieves better solutions than those of the existing algorithms.*
Keywords: Effective constraints, Feasibility statistics, Stochastic ranking evolutionary strategy, Engineering optimization problems

1. **Introduction.** Many real-world problems in scientific and engineering fields fall within the purview of *constrained optimization problems* (COPs) [1, 2, 3, 4]. Normally, those problems are composed of an objective function $f(\vec{x})$ and several equality and/or inequality constraints. In general, the mathematical formulation of COP has been defined as follows:

$$\text{minimize} \quad f(\vec{x}), \quad \vec{x} = (x_1, \ldots, x_n) \tag{1}$$
$$\text{subject to} \quad h_j(\vec{x}) = 0, \quad j = 1, 2, \ldots, q \tag{2}$$
$$g_j(\vec{x}) \le 0, \quad j = q+1, \ldots, m \tag{3}$$
$$\underline{x}_i \le x_i \le \overline{x}_i, \quad i = 1, \ldots, n$$

where n is the number of decision variables, $h_j(\vec{x})$ are q equality constraints, and $g_j(\vec{x})$ are $m-q$ inequality constraints. In COPs, each i-th variable is positioned between a lower boundary of $\underline{x}_i$ and an upper boundary of $\overline{x}_i$ such as $x_i \in [\underline{x}_i, \overline{x}_i]$.

During the last decade, many researchers have endeavored to solve COPs and developed several different types of algorithms. In particular, two novel heuristic algorithms, i.e., *evolutionary algorithms* (EAs) and *particle swarm optimization* (PSO), have attracted great attention in the latest years due to their search abilities for multiple solutions in parallel. To successfully solve COPs, both of them are amalgamated with several constraint-handling mechanisms which are enumerated below.

1) *Penalty function* approach modifies a COP into an unconstrained form by adding the degree of constraint violations as the penalty factors of r_j and c_j [5]. The penalty function $\phi(\vec{x})$ is formulated as follows:

$$\phi(\vec{x}) = f(\vec{x}) \pm \left[\sum_{j=1}^{q} r_j \times \mathcal{H}_j + \sum_{j=q+1}^{m} c_j \times \mathcal{G}_j \right], \tag{4}$$

where $\mathcal{H}_j$ and $\mathcal{G}_j$ are functions of the constraints that are indicated as $|h_j(\vec{x})|^\alpha$ and $\max\{0, g_j(\vec{x})\}^\beta$, in which the constants of α and β are set to 1 or 2, respectively. The principle and implementation are completely straightforward; however, it is quite difficult to discover suitable values of the penalty factors. In addition, this approach requires empirically fine-tuning a few system parameters. Some penalty functions based EAs have been introduced for solving COPs, effectively. In 2000, Coello [6] applied a notation of co-evolution to automatically discovering the penalty factors into genetic algorithms (GAs). Two years later, Hamida et al. [7] and Parsopoulos et al. [8] proposed an adaptive segregational constraint handling evolutionary algorithm (ASCHEA) and a non-stationary multi-stage assignment penalty function method based PSO, separately. Afterwards, Hu et al. [9, 10] introduced two PSO based constrained optimization algorithms in 2003 and 2007. The former algorithm was to preserve feasibility strategy for manipulating constraints, and the other employed the terminology of co-evolution for adopting adaptive penalty factors.

2) The second approach has taken into consideration of *separation between objective and constrained functions*. Runarsson et al. [11] proposed a stochastic ranking based evolutionary strategy (SRES) to balance the dominance between objective and constraint violations without considering the penalty factor in 2000. Next, Coello et al. [12] suggested a dominance-based selection scheme in which constraints were handled as additional objectives. Moreover, Montes et al. [13] in 2005 presented a simple multimembered evolutionary strategy (SMES) incorporated with a simple diversity mechanism that allowed infeasible solutions to remain in the population. Meanwhile, its separation approach was applied into PSO algorithms (as in He et al. [10]) in conjunction with a feasibility-based rule or binary tournament ranking selection (by Cagnian et al. [14]).

3) The third approach employs *special representations and operators* that simplify the shape of the search space to effectively conserve feasible solutions during the evolutions. At first, Kozei et al. [15] presented a special function called *homomorphous mapping* (HM) to transform the original problem into an unconstrained one using n-dimensional cube whose volume demonstrated feasible search spaces. Next, a gradient-based repair method (for GA) was introduced by [16]. It utilized the gradient information derived from each constraint to systematically convert infeasible solutions into feasible ones. In 2009, a new PSO algorithm using hybrid Nelder-Mead simplex search method (NM-PSO) was proposed by Zahara et al. [17]. It adopted two types of handling methods such as the gradient repair method and the constraint-fitness priority-based ranking method. Notwithstanding the flexibility of special operators, they are thoroughly problem-dependent and short of generality because of the complex constraints.

Moreover, they demand all feasible solutions at the initial stage, which in itself is also considerably hard to attain in many COPs.

4) *Hybrid approach* combines the mathematical or heuristic approaches such as random evolution, immune system, cultural algorithm and differential evolution [18, 19, 20, 21]. Until now, many hybrid algorithms have been developed for solving COPs.

Almost all of constrained optimization algorithms should test all constraints to decide the individual's feasibility; it leads to a large amount of the computational time and highly constrained feasible regions. When we consider a population of the i-th generation, many feasible solutions might exist. At that time, all constraints are not equally important. This implies that with those feasible individuals (of the population at the i-th generation), we can select a subset of meaningful constraints. In other words, there is a set of effective (i.e., hard) constraints with which the feasibility conditions can be satisfied.

In this paper, we propose a new constraint-handling technique that systematically discovers a set of effective constraints for the current (feasible) individuals, in which the feasibility conditions are satisfied at a sufficient level. In order to actually extract such constraints of COPs, we integrate the proposed constraint-handling technique into SRES, one of widely-known constrained evolutionary optimization algorithms [11]. Thanks to effective constraints made up of the smaller number of constraints than the originals, it yields a better search capability as preserving the same running time of SRES. For several comparative studies, we carry out computational simulations on three well-known engineering optimization problems between the proposed algorithm and several state-of-the-art references.

The rest of this paper is organized as follows. In Section 2, a brief introduction to SRES is stated. The proposed algorithm is presented in the following Section 3. Some test engineering optimization problems are presented in Section 4. In the following Section 5, preliminary experiments are carried out for supporting the validity of our approach. Comparative studies are performed in Section 6. This paper concludes with a brief summary in Section 7.

2. Stochastic Ranking Evolutionary Strategy (SRES).

Up to now, several constrained optimization evolutionary algorithms (COEAs) have been suggested. Among them, the stochastic ranking evolutionary strategy (SRES) [11], the simple multi-membered evolutionary strategy (SMES) [13] and the adaptive tradeoff model for evolutionary strategy (ATMES) [22], are well-known as evolutionary solvers for COPs.

The prior SRES adopted the stochastic ranking (SR) selection mechanism into the conventional evolutionary strategy (ES) to successfully manage the optimal solutions. Its ranking was simply implemented by the stochastic bubble sort procedure and could conventionally keep the balance of the dominance in a ranked set. The next SMES devised the diversity searching mechanisms such as the feasibility-based comparison mechanism, the initial step-size of the ES and the combined recombination, and the rest ATMES adaptively formulated three staged trade-off models with regard to the amount of feasible individuals of the population.

In this study, we address SRES as the evolutionary search algorithm, and combine it with our proposed constraint-handling technique (i.e., effective constraints) due to its uncomplicated search technique.

Initially, each individual of ES consists of two real-valued vectors such as design variables and step sizes $(\vec{x}, \vec{\sigma}) = \{(x_1, \cdots, x_n), (\sigma_1, \cdots, \sigma_n)\}$, where n is the number of decision variables in the given problem. At the initialization step, the first vector (i.e., design variables) is randomly generated by the parametric constraints of $\underline{x}_i \leq x_i \leq \overline{x}_i$,

Algorithm 1 *Pseudo code of stochastic ranking*

```
for i = 1 to λ do
    for j = 1 to λ − 1 do
        sample u ∈ U(0, 1), where U is a uniform distribution
        if (G(x_j) = G(x_{j+1} = 0) or (u < p_f) then
            if (f(x_j) > f(x_{j+1})) then
                swap(x_j, x_{j+1})
            end if
        else
            if (G(x_j) > G(x_{j+1})) then
                swap(x_j, x_{j+1})
            end if
        end if
    end for
    if no swap done then
        break
    end if
end for
```

where $\underline{x}_j$ is the lower bound and $\overline{x}_j$ is the upper bound, and the second vector (i.e., step sizes) is set to $(\overline{x}_j - \underline{x}_j)/\sqrt{n}$, $j = \{1, \cdots, n\}$, where n is the size of the design variables.

During the evolutionary procedures, the typical genetic operators of ES (viz., the global intermediate recombination and the Gaussian mutation) are applied into each population to explore the search space. The objective of the first operator is to create new strategy parameters of each offspring individual. Specifically, it is operated by stochastically selecting two parent individuals, and then its individual is calculated as an arithmetic mean of both parents, which is expressed as follows:

$$\hat{\sigma}_{h,j}^{(g)} = \left(\sigma_{i,j}^{(g)} + \sigma_{k,j}^{(g)}\right)/2, \tag{5}$$

where $h = \{1, \ldots, \lambda\}$, $i = \{1, \ldots, \mu\}$, $j = \{1, \ldots, n\}$ and k is a randomly chosen index from i (i.e., μ and λ are the size of parent and offspring, separately). Afterward, the mean step sizes are mutated by a log-normal update rule by the standard deviation, which is formulated as

$$\sigma_{h,j}^{(g+1)} = \hat{\sigma}_{h,j}^{(g)} \times \exp\left(\tau' N(0,1) + \tau N_j(0,1)\right), \tag{6}$$

where τ and τ' are learning rates set to $\varphi^*/\sqrt{2\sqrt{n}}$ and $\varphi^*/\sqrt{2n}$, respectively. The constant value of φ^* usually set to 1 is an expected rate of convergence, and $N(0, 1)$ is the normal distribution with zero mean and unit variance. Later, the design variables are mutated according to the modified probability density function characterized by Equation (7) to generate new offsprings. The mutated configuration is defined as follows:

$$x_{h,j}^{(g+1)} = x_{h,j}^{(g)} + \sigma_{h,j}^{(g+1)} \times N_j(0,1). \tag{7}$$

To balance the dominance between the objective and the constraint violations, it employs the stochastic ranking (SR) selection mechanism described in Algorithm 1. Through the SR algorithm, all individuals will be ranked based on the pair of objective and constraint violations $\left(f(\vec{x}_h), \mathcal{G}(\vec{x}_h)\right)$, where $\vec{x}_h$ denotes the solution of the h-th offspring individual, $h = \{1, \ldots, \lambda\}$ and $\mathcal{G}(\vec{x}_h)$ is the quadratic loss function $\sum_{j=1}^{m} \max\{0, g_j(\vec{x}_h)\}^2$ in term of g_j which is the j-th constraint. In this selection, a prefixed probability (p_f) should be properly determined to utilize only the objective function for comparing the ranks of

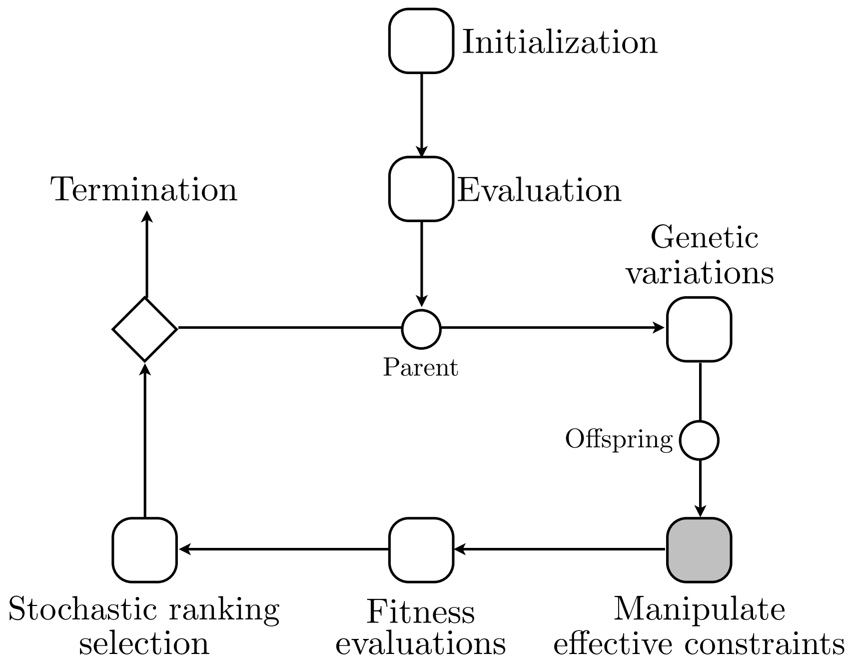


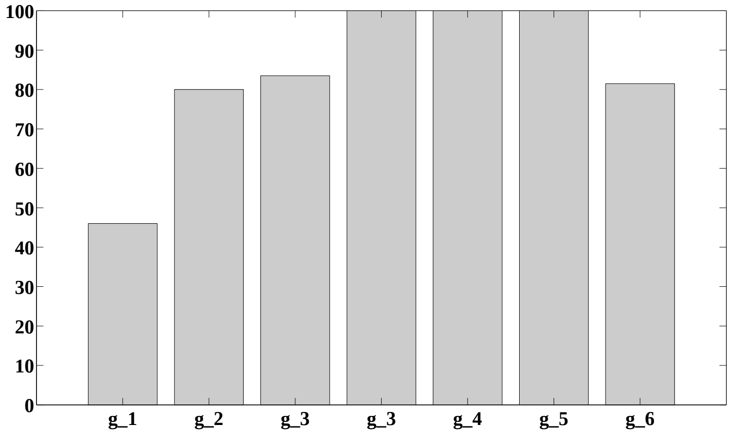
FIGURE 1. Diagram of the proposed *e*SRES

infeasible solutions [11]. In SRES, the evolutionary courses (i.e., evaluation, genetic operators and the SR selection) will be iterated until satisfying the stopping condition.

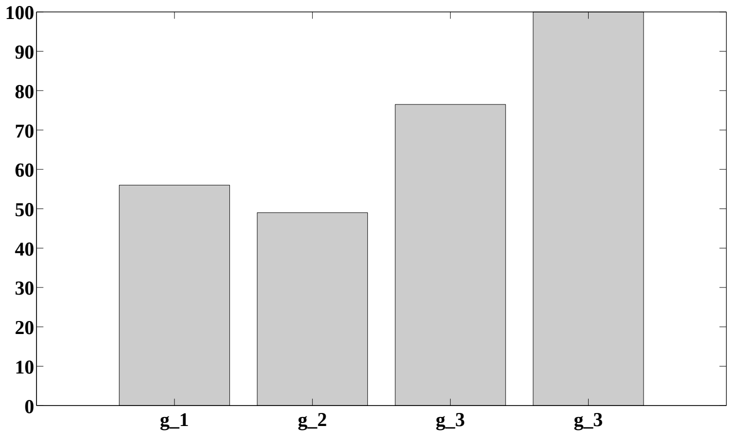
3. **The Proposed Constrained Evolutionary Algorithm: *e*SRES.** We present effective constraints based evolutionary algorithm incorporated with the configuration of SRES, which is termed *e*SRES. Its overall structure is illustrated in Figure 1. As a key feature, *e*SRES discovers effective constraints (for COPs) from the current individuals that retain the feasibility information. The motivation behind our approach is that all the constraints are not equally important when considering the current population (see Figure 2). Specifically, a set of some constraints is already sufficiently handled with the (current) individuals. In other words, it is adequate to employ the rest of the constraints for solving COPs with those individuals; such constraints are associated with *effective constraints*.

In relation to this, the proposed constraint-handling mechanism that methodically extracts the effective constraints based on the feasibility information of the individuals is described in detail. Beforehand, several control parameters should be defined for achieving our goal. However, it is very hard to determine their reasonable values since any prior knowledge for COPs is not generally available in advance. Thus, we pay attention to the fundamental knowledge of the given COP, such as the information on feasibility that can be obtained in testing constraint violations with the individuals. In other words, we define the *feasibility* of f_{g_j} as the number of feasible individuals in the population when considering the j-th constraint.

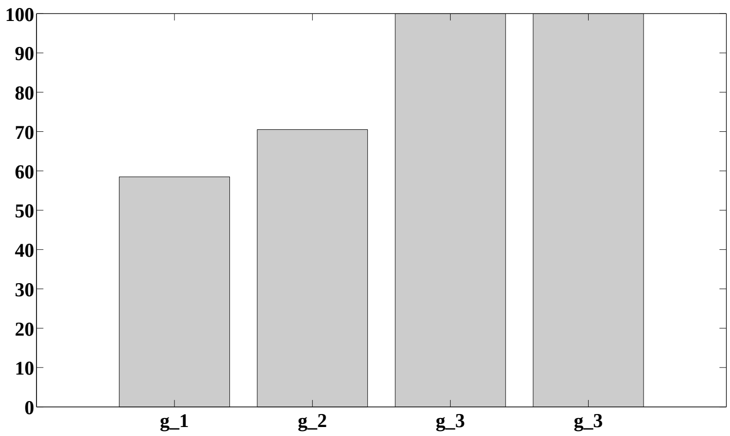
Based on the feasibility statistics, this study declares two system parameters of P_f and $P_{\mathcal{F}}$. The former is used for deciding the candidates of effective constraints in terms of each constraint. The latter offers a criterion whether a set of candidate constraints is accepted or not as a member of effective constraints in an average manner. The two parameters must be adaptively adjusted for the current individuals. Although it is impossible to determine their optimums of both measures, we empirically find out acceptable values by means of preliminary experiments with extensive setups in Section 5.



(a) Welded beam optimization problem



(b) Pressure vessel optimization problem



(c) Tension/compression spring optimization problem

Figure 2. The feasibility proportion with respect to each constraint on 100-th population of three engineering optimization problems

Incorporating with the two parameters, we count the number of feasible individuals at the current generation in order to build a set of effective constraints. If any feasible

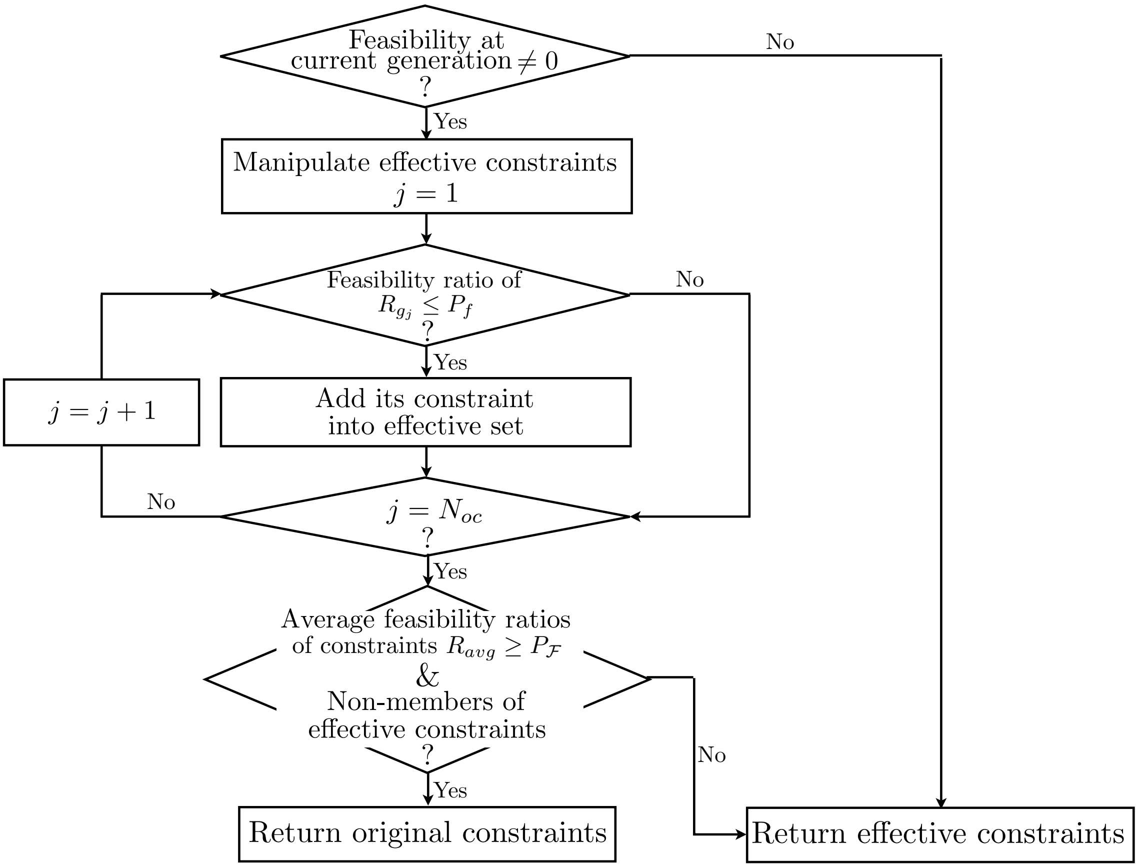


FIGURE 3. Details of the proposed constraint-handling mechanism for constructing effective constraints

solution exists in the population, the developed mechanism (see Figure 3) is run. Each candidate for effective constraints is assembled by comparing the prefixed P_f with the feasibility ratio defined by $R_{g_j} = f_{g_j}/\lambda \times 100\%$, where g_j denotes the j-th constraint, $j \in \{1, \ldots, N_{oc}\}$ (i.e., λ is the number of offspring, and N_{oc} is the number of original constraints). If the ratio of R_{g_j} is greater than the feasibility P_f, the j-th constraint is sufficiently covered with the current individuals; otherwise, the constraint can be included in the candidate set of effective constraints.

As the next step, we compute an average of feasibility ratios for all constraints by $R_{avg} = \frac{1}{N_{oc}} \sum_{j=1}^{N_{oc}} R_{g_j}$. In order to alleviate the effect of each feasibility on the overall constraints and avoid an empty set possibly taken place on the high level of feasibilities, the average R_{avg} is compared with the predefined $P_{\mathcal{F}}$. If $R_{avg} \geq P_{\mathcal{F}}$, we take the current constraints as a set of effective constraints.

For further understanding, we provide a detailed example when the proposed method is applied into the welded beam optimization problem [23] (see Figure 4). Prior to running eSRES, we set the two control parameters for feasibility; for instance, $P_f = 75\%$ and $P_{\mathcal{F}} = 50\%$. Initially, the number of the current feasible individuals is counted. As seen in the fourth figure, at the first generation, the proposed is invoked since some feasible solutions exist in the population. Thus, the feasibility ratio R_{g_j} (for each constraint) is compared with the prefixed $P_f(= 75\%)$. There are only three constraints whose R_{g_j} are smaller than P_f; $R_{g_2} = 54.5\%$, $R_{g_3} = 48.5\%$ and $R_{g_4} = 59.5\%$. It denotes that the three constraints are regarded as the potential effective constraints. Afterwards, an average of feasibility ratios is computed; in this example, $R_{avg} = 73.57\%$. The average is then

	Feasibility (%)	Feasible ratio of constraints R_{g_1}	R_{g_2}	R_{g_3}	R_{g_4}	R_{g_5}	R_{g_6}	R_{g_7}	R_{avg} (%)	Effective constaints (E)
Generation#1	2.5%	84.0%	54.5%	48.5%	59.5%	98.0%	81.5%	89.0%	73.57%	$E = [g_2, g_3, g_4]$
Generation#2	3.0%	80.5%	57.5%	52.5%	60.5%	98.5%	86.0%	94.0%	75.64%	$E = [g_2, g_3, g_4]$
Generation#3	7.5%	70.5%	61.0%	51.0%	71.0%	97.5%	89.5%	93.5%	76.29%	$E = [g_1, g_2, g_3, g_4]$
Generation#4	6.0%	85.0%	65.5%	42.5%	64.5%	100%	92.0%	89.0%	76.93%	$E = [g_2, g_3, g_4]$
Generation#5	11.0%	74.5%	64.5%	57.5%	69.5%	97.0%	87.5%	93.0%	77.64%	$E = [g_1, g_2, g_3, g_4]$
⋮	$R_{g_j} \leq P_f = 75\%$ $R_{avg} \geq P_{\mathcal{F}} = 50\%$									

FIGURE 4. Example of extracting the effective constraints on the welded beam optimization problem

compared with another predefined parameter $P_{\mathcal{F}}(= 50\%)$. Since R_{avg} is greater than $P_{\mathcal{F}}$, the three constraints, g_2, g_3 and g_4, are decided as the effective constraints. Thus, a set of effective constraints such as $E = [g_2, g_3, g_4]$ is discovered at the first generation. Similarly, we can obtain the effective constraints $E = [g_2, g_3, g_4]$, $E = [g_1, g_2, g_3, g_4]$ and $E = [g_2, g_3, g_4]$ at the subsequent generations.

Because of the incorporation of effective constraints, the proposed *e*SRES can discover the global optimum with the smaller number of constraints than the originals. In other words, the population can be evolved toward the legitimate region of the search space without estimating all constraints.

4. **Engineering Optimization Problems.**

4.1. **Welded-beam optimization problem: E01.**

This problem [23] is to minimize the cost of a welded beam subject to seven constrained functions formulated by shear stress (τ), bending stress in the beam (σ), buckling load on the bar (p_c), end deflection of the beam (δ) and side constraints. The mathematical forms of constraints are stated in Equations (9) and (10). In this problem, the optimal cost of the objective function will be achieved by four design variables whose intervals are set to $0.1 \leq x_1,\ x_4 \leq 2.0$ and $0.1 \leq x_2,\ x_3 \leq 10$, respectively.

$$\text{minimize} \quad f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \tag{8}$$

$$\text{subject to} \tag{9}$$

$$\begin{aligned}
g_1(\vec{x}) &= \tau - 13600 \leq 0 \\
g_2(\vec{x}) &= \sigma - 30000 \\
g_3(\vec{x}) &= x_1 - x_4 \leq 0 \\
g_4(\vec{x}) &= 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5.0 \leq 0 \\
g_5(\vec{x}) &= 0.125 - x_1 \leq 0 \\
g_6(\vec{x}) &= \delta - 0.25 \leq 0 \\
g_7(\vec{x}) &= 6000 - pc \leq 0,
\end{aligned}$$

where

$$p_c = \frac{4.013(30\times 10^6)\sqrt{\frac{x_3^2 x_4^6}{36}}}{196}\left(1-\frac{x_3\sqrt{\frac{30\times 10^6}{4(12\times 10^6)}}}{28}\right); \quad \delta = \frac{65856000}{(30\times 10^6)x_3^3 x_4}; \tag{10}$$

$$\sigma = \frac{504000}{x_3^2 x_4}; \quad J = 2\left\{x_1 x_2\sqrt{2}\left[\frac{x_2^2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\}; \quad R = \sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2};$$

$$M = 6000\left(14+\frac{x_2}{2}\right); \quad \tau = \sqrt{t_1^2+2(t_1 t_2)\frac{x_2}{2R}+t_2^2}; \quad t_1 = \frac{6000}{\sqrt{2}x_1 x_2}; \quad t_2 = \frac{MR}{J}.$$

4.2. **Pressure vessel optimization problem: E02.** This engineering optimization problem [24] is to acquire a total minimum cost of the material, forming and welding in accordance with thickness of the shall ($0 \le x_1 \le 99$), thickness of the head ($0 \le x_2 \le 99$), inner radius ($10 \le x_3 \le 200$) and length of the cylindrical section of the vessel ($10 \le x_4 \le 200$) subject to four inequality constrained functions which are expressed in Equation (12).

minimize (11)

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

subject to (12)

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0$$
$$g_3(\vec{x}) = \pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$
$$g_4(\vec{x}) = x_4 - 240 \le 0.$$

4.3. **Tension/compression spring optimization problem: E03.** The aim of this problem [25] is to minimize the weight of a tension/compression spring subject to four constraints in accordance with deflection, shear stress, surge frequency and limits on outside diameter which are specified in Equation (14). In this optimization problem, three design variables are defined as follows: the wire diameter of $0.05 \le x_1 \le 2.0$, the mean coil diameter of $0.25 \le x_2 \le 1.3$ and the number of dominant coils of $2.0 \le x_3 \le 15.0$.

minimize (13)

$$f(\vec{x}) = (x_3+2)x_1^2 x_2$$

subject to (14)

$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \le 0$$
$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_1^3x_2 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \le 0$$
$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2 x_3} \le 0$$
$$g_4(\vec{x}) = \frac{x_1+x_2}{1.5} - 1 \le 0.$$

5. **Preliminary Experiments.** In this section, we carry out numerical simulations to determine the justifiable proportions of two feasibility parameters (i.e., P_f and $P_{\mathcal{F}}$) with regard to a variety of different setups. During all experiments, the size of population and

TABLE 1. The statistical results of welded beam optimization problem with different control feasibility parameters $P_{\mathcal{F}}$ and P_f

		$P_{\mathcal{F}} = 50\%$	$P_{\mathcal{F}} = 55\%$	$P_{\mathcal{F}} = 60\%$	$P_{\mathcal{F}} = 65\%$	$P_{\mathcal{F}} = 70\%$
$P_f = 60\%$	mean	1.724864	1.724872	1.724853	1.724865	1.724853
	st. dev.	5.96.E-05	6.46.E-05	6.46.E-07	5.02.E-05	5.40.E-06
	frequency	114.20	102.57	102.37	105.00	89.30
$P_f = 65\%$	mean	1.728321	1.727554	1.728076	1.728765	1.725480
	st. dev.	7.36.E-03	7.27.E-03	8.56.E-03	1.14.E-02	1.98.E-03
	frequency	242.13	235.10	223.60	239.60	206.63
$P_f = 70\%$	mean	1.725091	1.725158	1.725191	1.725218	1.724966
	st. dev.	6.12.E-04	6.00.E-04	8.51.E-04	6.07.E-04	2.04.E-04
	frequency	278.30	278.03	278.03	279.63	271.10
$P_f = 75\%$	mean	**1.724852**	1.724852	1.724852	1.724852	1.724852
	st. dev.	**2.51.E-07**	3.02.E-08	2.20.E-08	4.15.E-08	1.93.E-08
	frequency	**293.00**	293.60	292.43	293.33	291.63
$P_f = 80\%$	mean	1.724852	1.724852	1.724852	1.724852	1.724852
	st. dev.	5.00.E-12	6.90.E-11	7.00.E-13	1.70.E-12	1.80.E-12
	frequency	299.13	298.97	298.87	298.87	298.90
$P_f = 85\%$	mean	1.724852	1.724852	1.724852	1.724852	1.724852
	st. dev.	1.00.E-13	0.00.E+00	0.00.E+00	0.00.E+00	1.00.E-13
	frequency	293.43	295.87	296.07	289.20	293.43
$P_f = 90\%$	mean	1.724852	1.724852	1.724852	1.724852	1.724865
	st. dev.	0.00.E+00	0.00.E+00	0.00.E+00	0.00.E+00	7.14.E-05
	frequency	292.17	292.30	293.00	290.37	291.70
$P_f = 95\%$	mean	1.724852	1.724864	1.724852	1.724852	1.724852
	st. dev.	0.00.E+00	6.42.E-05	0.00.E+00	0.00.E+00	0.00.E+00
	frequency	299.43	299.47	299.40	299.23	299.47

the maximum number of generations of the proposed eSRES are set to $(\mu, \lambda) = (30, 200)$ and 300, respectively.

In every computational tests, the first parameter is categorized as $P_f = \{60\%, 65\%, 70\%, 75\%, 80\%, 85\%, 90\%, 95\%\}$ for taking account of the diverse conditions. We only assert more than $P_f = 60\%$ since the less than its degree may make all constraints effective constraints. The rest parameter of $P_{\mathcal{F}}$ is arranged by five demonstrations such as $P_{\mathcal{F}} = \{50\%, 55\%, 60\%, 65\%, 70\%\}$. Here, we do not study the under $P_{\mathcal{F}} = 50\%$ since the premature convergence may take place during those scopes. Also, we set aside the over $P_{\mathcal{F}} = 70\%$ whose statements are equal to the conventional SRES. The proposed eSRES is executed with the above diverse instances to adopt the reasonable values of feasibility parameters through three kinds of engineering optimization problems.

There are three computational experiments in accordance with welded beam optimization problem (see Table 1), pressure vessel optimization problem (see Table 2) and tension/compression spring optimization problem (see Table 3). They display the statistics of the discovered optimal solutions and the average occurrence of effective constraints during all generations over 30 independent runs.

Firstly, we make an assessment of one system parameter $P_{\mathcal{F}}$ based on three kinds of statistical results. Each instance of $P_{\mathcal{F}}$ discovered highly comparable values, so we settle on the smallest proportion of $P_{\mathcal{F}} = 50\%$ to manipulate the proposed constraint-handling approach in the earlier generation.

Table 2. The statistical results of pressure vessel optimization problem with different control feasibility parameters $P_{\mathcal{F}}$ and P_f

		$P_{\mathcal{F}} = 50\%$	$P_{\mathcal{F}} = 55\%$	$P_{\mathcal{F}} = 60\%$	$P_{\mathcal{F}} = 65\%$	$P_{\mathcal{F}} = 70\%$
$P_f = 60\%$	mean	5958.5767	5998.2914	5956.1957	5974.0309	5926.6933
	st. dev.	6.88.E+01	1.40.E+02	5.62.E+01	1.52.E+02	5.13.E+01
	frequency	202.47	199.73	170.60	130.17	49.27
$P_f = 65\%$	mean	6114.4619	6022.8775	6187.8112	6055.2301	5984.2739
	st. dev.	1.83.E+02	1.92.E+02	2.06.E+02	1.77.E+02	1.91.E+02
	frequency	258.83	259.63	259.80	221.70	132.73
$P_f = 70\%$	mean	5931.9133	5919.1503	5912.7209	5938.8408	5900.8776
	st. dev.	1.23.E+02	4.11.E+01	3.27.E+01	1.06.E+02	2.22.E+01
	frequency	293.13	292.40	289.87	264.23	192.63
$P_f = 75\%$	mean	**5892.8034**	5894.6429	5890.5004	5890.4138	5896.3717
	st. dev.	**1.57.E+01**	1.61.E+01	8.66.E+00	8.78.E+00	1.67.E+01
	frequency	**297.90**	298.67	297.90	288.80	246.07
$P_f = 80\%$	mean	5917.0025	5917.9391	5920.8288	5918.1855	5913.6057
	st. dev.	3.37.E+01	3.04.E+01	2.95.E+01	3.40.E+01	4.27.E+01
	frequency	299.33	299.60	299.50	298.87	289.93
$P_f = 85\%$	mean	5923.7388	5926.8564	5961.0431	5939.6900	5930.0067
	st. dev.	3.42.E+01	5.36.E+01	1.14.E+02	6.32.E+01	4.13.E+01
	frequency	291.97	291.10	291.43	292.57	292.03
$P_f = 90\%$	mean	5929.4364	5934.0554	5947.3431	5948.6963	5948.3426
	st. dev.	5.07.E+01	6.04.E+01	6.34.E+01	6.74.E+01	8.20.E+01
	frequency	294.27	293.47	294.43	295.40	293.70
$P_f = 95\%$	mean	5940.9751	5979.4995	5952.1605	5923.0507	5938.5898
	st. dev.	6.91.E+01	9.94.E+01	6.05.E+01	3.88.E+01	6.79.E+01
	frequency	298.90	299.43	299.63	299.10	299.47

According to the above prefixed $P_{\mathcal{F}}(= 50\%)$, we study the remaining parameter of P_f. In the first simulation result shown in Table 1, five cases such as $P_f = \{75\%, 80\%, 85\%, 90\%, 95\%\}$ found completely comparable statistical values. The next table showed that every instance indicated similar optimum and the mean of frequencies for effective constraints. Among them, $P_f = 75\%$ designated the best searching quality with the smallest mean and standard deviation values. At the last numerical statistics given in Table 3, all occasions discovered the considerable approximations. To sum up three computational analyses, we ultimately determined that the reasonable proportions of two control parameters were set to $P_{\mathcal{F}} = 50\%$ and $P_f = 75\%$, respectively.

To study the influence of the proposed constraint-handling approach of effective constraints, we adopt two kinds of SRES with the different sets of constraints. The first algorithm utilizes all constraints like the original SRES, and the rest engages the discovered high frequency of effective constraints by the proposed *e*SRES which is termed as *d*SRES. There are the found meaningful constraints such as $E_{E01} = [g_1, g_2, g_3, g_7]$, $E_{E02} = [g_1, g_2, g_3]$ and $E_{E03} = [g_1, g_2]$.

At first, we execute the widely known statistical test of t-test with two samples as regarding three instances such as *e*SRES vs. SRES, *e*SRES vs. *d*SRES and SRES vs. *d*SRES. Table 4 shows p-values of the tests with respect to each case. Each value is the probability observing a value as extreme or more extreme of the test statistic. If its value is close to 1, two samples are quite similar. Also, we compare the optimal solutions from

TABLE 3. The statistical results of tension/compression spring optimization problem with different control feasibility parameters $P_{\mathcal{F}}$ and P_f

		$P_{\mathcal{F}} = 50\%$	$P_{\mathcal{F}} = 55\%$	$P_{\mathcal{F}} = 60\%$	$P_{\mathcal{F}} = 65\%$	$P_{\mathcal{F}} = 70\%$
$P_f = 60\%$	mean	0.0099384	0.0099319	0.0099507	0.0099407	0.0099705
	st. dev.	9.33.E-05	9.36.E-05	1.27.E-04	1.16.E-04	1.35.E-04
	frequency	267.60	271.00	270.57	261.83	268.80
$P_f = 65\%$	mean	0.0098734	0.0098725	0.0098725	0.0098725	0.0098729
	st. dev.	5.08.E-06	3.56.E-11	5.00.E-11	1.78.E-09	2.35.E-06
	frequency	283.77	286.00	285.87	282.07	280.17
$P_f = 70\%$	mean	0.0098751	0.0098783	0.0098773	0.0098749	0.0098748
	st. dev.	8.57.E-06	1.54.E-05	1.37.E-05	1.02.E-05	8.64.E-06
	frequency	292.23	293.97	289.63	290.20	288.30
$P_f = 75\%$	mean	**0.0098798**	0.0099104	0.0098811	0.0099046	0.0098781
	st. dev.	**1.75.E-05**	1.01.E-04	3.01.E-05	1.43.E-04	2.08.E-05
	frequency	**264.73**	266.93	263.60	262.77	261.20
$P_f = 80\%$	mean	0.0098744	0.0098823	0.0099113	0.0098775	0.0098735
	st. dev.	7.90.E-06	3.23.E-05	1.15.E-04	1.88.E-05	3.56.E-06
	frequency	252.47	249.73	256.40	250.30	230.47
$P_f = 85\%$	mean	0.0098771	0.0098838	0.0098847	0.0098821	0.0098786
	st. dev.	1.55.E-05	4.25.E-05	4.31.E-05	2.50.E-05	2.08.E-05
	frequency	276.93	270.93	272.97	272.80	270.00
$P_f = 90\%$	mean	0.0099135	0.0098732	0.0098737	0.0098793	0.0098800
	st. dev.	1.17E-04	2.53E-06	6.62E-06	3.70E-05	2.79E-05
	frequency	292.47	292.50	291.03	291.50	290.47
$P_f = 95\%$	mean	0.0098782	0.0098911	0.0098750	0.0098804	0.0098984
	st. dev.	2.34.E-05	6.86.E-05	8.24.E-06	2.62.E-05	9.28.E-05
	frequency	299.03	299.20	298.50	297.03	294.67

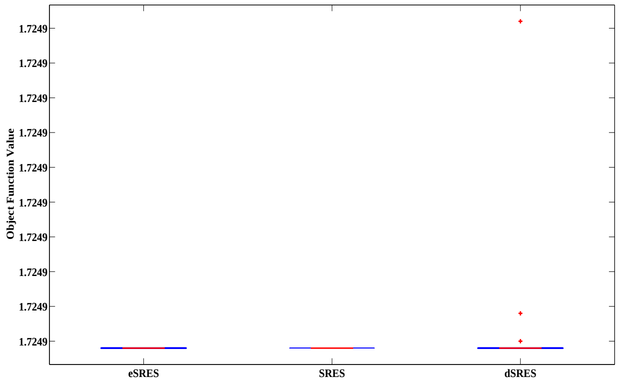
TABLE 4. t-Test. The values represent p-values

	E01	E02	E03
*e*SRES vs. SRES	1.000000.E+00	1.405192.E-04	1.007997.E-01
*e*SRES vs. *d*SRES	2.693650.E-01	4.029003.E-07	2.544033.E-02
SRES vs. *d*SRES	2.693650.E-01	6.696392.E-05	1.437815.E-01

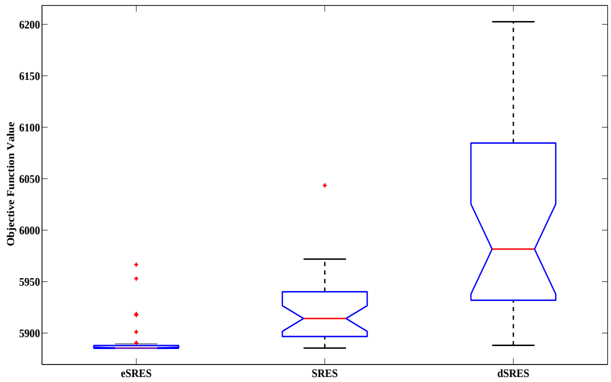
three algorithms over 30 independent runs on three engineering problems, as illustrated in Figure 5. Those figures display that the proposed *e*SRES discovers the better optimal solutions with small distribution than those of the others. Through statistical tests, we are able to investigate the dominance of our algorithm.

Next, we evaluate the computing time of *e*SRES and SRES over 30 independent runs to investigate the efficiency of the proposed constraint-handling technique (i.e., effective constraints). Table 5 describes the statistical comparisons of the running time in accordance with three engineering optimization problems. In all cases, the proposed *e*SRES devoted the remarkably competitive computational time with SRES. Throughout the above simulation results, we could validate the influence of the proposed effective constraints.

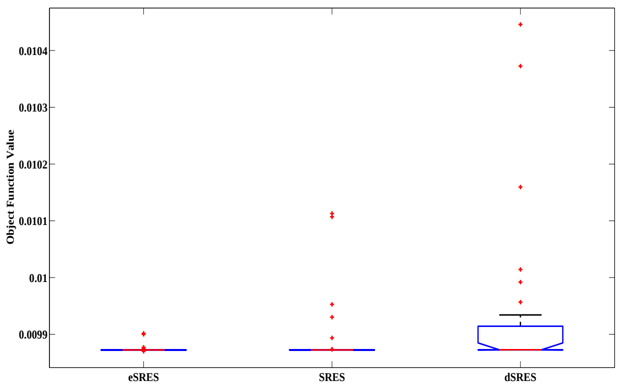
6. **Computational Studies.** In this section, we experiment numerical tests on three engineering optimization problems such as welded beam problem (Section 6.1), pressure



(a) Welded beam optimization problem



(b) Pressure vessel optimization problem



(c) Tension/compression spring optimization problem

FIGURE 5. The comparison of the optimal solutions from *e*SRES, SRES and *d*SRES on three engineering optimization problems over 30 independent runs

vessel problem (Section 6.2) and tension/compression spring problem (Section 6.3) for investigating the performances of our algorithm. In whole simulation results, we compare the proposed *e*SRES with well-known references which are summarized as follows:

TABLE 5. Comparing the computational time between *e*SRES and SRES

		*e*SRES	SRES [11]
E01	min	1.12 sec	1.13 sec
	mean	1.14 sec	1.16 sec
	max	1.18 sec	1.25 sec
	st. dev.	1.50.E-02 sec	2.68.E-02 sec
E02	min	1.07 sec	1.08 sec
	mean	1.09 sec	1.09 sec
	max	1.16 sec	1.18 sec
	st. dev.	1.66.E-02 sec	1.79.E-02 sec
E03	min	1.09 sec	1.09 sec
	mean	1.11 sec	1.13 sec
	max	1.20 sec	1.37 sec
	st. dev.	1.94.E-02 sec	5.11.E-02 sec

- GA1 [6] utilized co-evolution mechanism for automatically adjusting penalty factors of a fitness function. This constraint-handling method was incorporated with a genetic algorithm (GA).
- GA2 [12] proposed the multi-objective optimization (MOO) techniques based method by means of considering the separation between objective and constraint violations for overcoming the drawback of the penalty function method (i.e., the empirically fine-tuning parameters).
- HE-PSO [26] suggested a new particle swarm optimization (PSO) for solving COPs as adopting the death penalty mechanism which did not use all infeasible solutions during whole procedures.
- CPSO [9] proposed co-evolution based PSO algorithm for handling decision solutions and constraints. The aim of this algorithm was to search the optimal solutions and penalty factors.
- HPSO [10] employed the feasibility-based rule to manage constraints without additional parameters and to guide the particles into the feasible region, quickly. In addition, a simulated annealing (SA) was applied into the best solution for avoiding the premature convergence.
- NM-PSO [17] integrated the Nelder-Mead (NM) simplex search method with PSO algorithm. This algorithm took on the special operators, i.e., the gradient repair method and the constraint fitness priority-based ranking, to renovate infeasible solutions into feasible ones, effectively.

For each test problem, our algorithm has been performed during 30 independent runs. The size of population is set to (μ: parent, λ: offspring) = $(30, 200)$, and the number of maximum generations is fixed to 300; hence, the number of of the fitness function evaluations (FFEs) are computed by $60,000$.

The following subsections represent the statistical results of the proposed algorithm *e*SRES and six well-known constrained optimization algorithms that have been reported in the special literature with respect to three engineering optimization problems. In Table 6, we describe the FFEs of the referred algorithms.

6.1. **Simulation results of welded-beam optimization problem.** This problem was taken from [23], and its aim was to minimize the cost of the welded beam subject to seven constraints. Table 7 lists the optimal solutions which are obtained by the above mentioned

Table 6. Fitness function evaluations of the compared algorithms, where N is the number of design variables in the given COP

	Fitness function evaluations
GA1 [6]	$900,000$
GA2 [12]	$80,000$
HE-PSO [26]	$200,000$
CPSO [9]	$200,000$
HPSO [10]	$81,000$
NM-PSO [17]	$(21N + 1) \times 1000$
eSRES	$60,000$

Table 7. Comparison of the best solution on welded beam optimization problem

Algorithms	x_1	x_2	x_3	x_4	$f(\vec{x})$
GA1 [6]	0.208800	3.420500	8.997500	0.210000	1.748309
GA2 [12]	0.205986	3.471328	9.020224	0.206480	1.728226
HE-PSO [26]	0.20573	3.47049	9.03662	0.20573	1.724851
CPSO [9]	0.202369	3.544214	9.048210	0.205723	1.728024
HPSO [10]	0.235730	3.470489	9.036624	0.205730	1.724852
NM-PSO [17]	0.205830	3.468338	9.036624	0.205730	1.724717
eSRES	0.205730	3.470489	9.036624	0.205730	1.724852

Table 8. Statistical results for welded beam optimization problem

Algorithms	best	mean	worst	st. dev.
GA1 [6]	1.748309	1.771973	1.785835	1.12.E-02
GA2 [12]	1.728226	1.792654	1.993408	7.47.E-02
HE-PSO [26]	1.724851	1.724852	NA	0.00.E+00
CPSO [9]	1.728024	1.748831	1.782143	1.29.E-02
HPSO [10]	1.724852	1.749040	1.814295	4.00.E-02
NM-PSO [17]	1.724717	1.726373	1.733393	3.50.E-03
eSRES	1.724852	1.724852	1.724852	2.51.E-07

optimization algorithms as well as eSRES, and their statistical simulation results are shown in Table 8.

From Table 8, it could be seen that the best feasible solution was obtained by NM-PSO, but our proposed algorithm discovered the quite comparative solution. In the next table, we confirmed that the average searching quality of eSRES was much better than those of other algorithms since our algorithm had the smallest the mean and standard deviation in the problem.

6.2. **Simulation results of pressure vessel optimization problem.** Its optimization problem was introduced by Kannan et al. in 1994 to minimize the total cost of material, forming and welding [24]. The best solution obtained by the references and eSRES are reported in Table 9, and the statistics of the solutions are compiled in Table 10.

It was described in Table 9 that the best feasible solution obtained by our algorithm was competitive to the result of references. From Table 10, it could be discovered that the

TABLE 9. Comparison of the best solution on pressure vessel optimization problem

Algorithms	x_1	x_2	x_3	x_4	$f(\vec{x})$
GA1 [6]	0.8125	0.4375	40.3239	200.0000	6288.7445
GA2 [12]	0.8215	0.4375	42.0974	176.6540	6059.9463
HE-PSO [26]	0.8125	0.4375	42.09845	176.6366	6059.1312
CPSO [9]	0.8125	0.4375	42.0913	176.7465	6061.0777
HPSO [10]	0.8125	0.4375	42.0984	176.6366	6059.7143
NM-PSO [17]	0.8036	0.3972	41.6392	182.4120	5930.3137
*e*SRES	0.7782	0.3846	40.3197	199.9991	5885.3359

TABLE 10. Statistical results for pressure vessel optimization problem

Algorithms	best	mean	worst	st. dev.
GA1 [6]	6288.7445	6293.8432	6308.1497	7.41.E+00
GA2 [12]	6059.9463	6177.2533	6469.3220	1.31.E+02
HE-PSO [26]	6059.1313	NA	NA	NA
CPSO [9]	6061.0777	6147.1332	6363.8041	8.65.E+01
HPSO [10]	6059.7143	6099.9323	6288.6770	8.62.E+01
NM-PSO [17]	5930.3140	5946.7900	5960.0560	9.16.E+00
*e*SRES	5885.3359	5892.8034	5939.9787	1.57.E+01

TABLE 11. Comparison of the best solution on tension/compression spring optimization problem

Algorithms	x_1	x_2	x_3	$f(\vec{x})$
GA1 [6]	0.051480	0.351661	11.632201	0.0127048
GA2 [12]	0.051989	0.363965	10.890522	0.0126810
HE-PSO [26]	0.051466	0.351384	11.608659	0.0126662
CPSO [9]	0.051728	0.357644	11.244543	0.0126747
HPSO [10]	0.051706	0.357126	11.265083	0.0126652
NM-PSO [17]	0.051620	0.355498	11.333272	0.0126302
*e*SRES	0.050000	0.374433	8.546569	0.0098725

high-quality searching capacity of *e*SRES to the other referred optimization algorithms with the better solutions in accordance with every case.

6.3. **Simulation results of tension/compression spring optimization problem.** This optimization problem was to minimize the weight of the tension spring subject to four inequality constraints, introduced by *Arora* in 1989 [25]. Table 11 and Table 12 present the optimal solutions obtained by *e*SRES as well as our references and their statistical results, respectively.

It was shown in Table 12 that the performances of *e*SRES were even better than those of the our compared algorithms, and the worst solution of ours was smaller than the optimal values of the other referred algorithms in the problem.

To sum up those experimental results and comparisons of the above three engineering optimization problems, we could verify the superiority of the proposed *e*SRES.

TABLE 12. Statistical results for tension/compression spring optimization problem

Algorithms	best	mean	worst	st. dev.
GA1 [6]	0.0127048	0.0127690	0.0128220	3.34.E-05
GA2 [12]	0.0126810	0.0127420	0.0129730	5.90.E-05
HE-PSO [26]	0.012666	0.012719	NA	6.45.E-05
CPSO [9]	0.0126747	0.012730	0.012924	5.20.E-04
HPSO [10]	0.0126652	0.0127072	0.0127191	1.58.E-05
NM-PSO [17]	0.0126302	0.0126314	0.012633	8.74.E-07
*e*SRES	0.0098725	0.0098798	0.009949	1.75.E-05

7. **Conclusion.** This paper has presented the new constraint handling method for solving COPs, efficiently. The aim of our proposed approach was to extract the actual subset of meaningful constraints (i.e., *effective constraints*). Due to its constraints, we were able to evolve the population toward the reasonable feasible regions without examining entire constraints. To achieve effective constraints, we have newly designed the intelligible mechanism using feasible statistics of the current generation and two predefined feasibility parameters whose justifiable levels were specified by preliminary studies. From the numerical simulations and comparisons of three engineering optimization problems, we were assured that the proposed *e*SRES could discover the better optimal solution than those of well-known references.

Our devised technique for obtaining effective constraints was quite straightforward, so it will be comfortably applied into the other constrained optimization algorithms. However, we require that the prefixed feasibility proportions should be modified for a variety of COPs. That is our future search issue.

Acknowledgements. This work was financially supported in part by "Unmanned Technology Research Center at KAIST, originally funded by DAPA, ADD, Korea" and "GIST Systems Biology Infrastructure Establishment Grant, Korea", and also the authors would like to thank Dr. Yaochu Jin and anonymous reviewers for useful discussions and comments.

REFERENCES

[1] Y. Guo, X. Cao and J. Zhang, Constraint handling based multiobjective evolutionary algorithm for aircraft landing scheduling, *International Journal of Innovative Computing, Information and Control*, vol.5, no.8, pp.2229-2238, 2009.

[2] T. Uno, H. Katagiri and K. Kato, An evolutionary multi-agent based search method for stackelberg solutions of bilevel facility location problems, *International Journal of Innovative Computing, Information and Control*, vol.4, no.5, pp.1033-1042, 2008.

[3] C. Liu, An evolutionary algorithm for solving dynamic nonlinear constrained optimization, *ICIC Express Letters*, vol.4, no.3(B), pp.1039-1044, 2010.

[4] S. Oh, Y. Jin and M. Jeon, Approximate models for constraint functions in evolutionary constrained optimization, *International Journal of Innovative Computing, Information and Control*, vol.7, no.11, pp.6585-6603, 2011.

[5] C. A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering*, vol.191, no.11-12, pp.1245-1287, 2002.

[6] C. A. Coello Coello, Use of a self-adaptive penalty approach for engineering optimization problems, *Computers in Industry*, vol.41, no.2, pp.113-127, 2000.

[7] S. B. Hamida and M. Schoenauer, ASCHEA: New results using adaptive segregational constraint handling, *Proc. of IEEE Conference on Evolutionary Computation 2002*, Honolulu, HI, USA, pp.82-87, 2002.

[8] K. E. Parsopoulos and M. N. Vrahatis, Particle swarm optimization method for constrained optimization problems, *Proc. of the Euro-International Symposium on Computational Intelligence 2002*, Košice, Slovakia, pp.214-220, 2002.
[9] Q. He and L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, *Engineering Applications of Artificial Intelligence*, vol.20, no.1, pp.89-99, 2007.
[10] Q. He and L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, *Applied Mathematics and Computation*, vol.186, no.2, pp.1407-1422, 2007.
[11] T. P. Runarsson and X. Yao, Stochastic ranking for constrained evolutionary optimization, *IEEE Transactions on Evolutionary Computation*, vol.4, no.3, pp.284-294, 2000.
[12] C. A. Coello Coello and E. M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, *Advanced Engineering Informatics*, vol.16, no.3, pp.193-203, 2002.
[13] E. M. Montes and C. A. Coello Coello, A simple multimembered evolution strategy to solve constrained optimization problems, *IEEE Transactions on Evolutionary Computation*, vol.9, no.1, pp.1-17, 2005.
[14] L. C. Cagnina, S. C. Esquivel and C. A. Coello Coello, Solving engineering optimization problems with the simple constrained particle swarm optimizer, *Informatica*, vol.32, pp.319-326, 2008.
[15] S. Koziel and Z. Michalewicz, Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization, *Evolutionary Computation*, vol.7, no.1, pp.19-44, 1999.
[16] P. Chootinan and A. Chen, Constraint handling in genetic algorithms using a gradient-based repair method, *Computer and Operations Research*, vol.33, pp.2263-2281, 2006.
[17] E. Zahara and Y.-T. Kao, Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems, *Expert Systems with Applications*, vol.36, no.2, pp.3880-3886, 2009.
[18] S. V. Belur, CORE: Constrained optimization by random evolution, *Late Breaking Papers at the Genetic Programming 1997 Conference*, CA, USA, 1997.
[19] S. Forrest and A. S. Perelson, Genetic algorithms and the immune system, *Proc. of the 1st Workshop on Parallel Problem Solving from Nature*, Berlin, Germany, pp.320-325, 1991.
[20] R. G. Reynolds, An introduction to cultural algorithms, *Proc. of the 3rd Annual Conference on Evolutionary Programming*, pp.131-134, 1994.
[21] X. Zhang, Q. Lu, S. Wen, M. Wu and X. Wang, A modified differential evolution for constrained optimization, *ICIC Express Letters*, vol.2, no.2, pp.181-186, 2008.
[22] Y. Wang, Z. Cai, Y. Zhou and W. Zeng, An adaptive tradeoff model for constrained evolutionary optimization, *IEEE Transactions on Evolutionary Computation*, vol.12, no.1, pp.80-92, 2008.
[23] K. Ragsdell and D. Phillips, Optimaldesign of a class of welded structures using geometric programming, *Journal of Engineering for Industry*, vol.98, no.3, pp.1021-1025, 1976.
[24] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, *ASME Journal of Mechanical Design*, vol.122, no.2, pp.223-229, 1990.
[25] A. D. Belegundu, *A Study of Mathematical Programming Methods for Structure Optimization*, Ph.D. Thesis, University of Iowa, 1982.
[26] X. Hu, R. C. Eberhart and Y. Shi, Engineering optimization with particle swarm, *Proc. of the IEEE Swarm Intelligence Symposium 2003*, Indianapolis, IN, USA, pp.53-57, 2003.