Constraint Handling in Multiobjective Evolutionary Optimization

Yonas Gebre Woldesenbet, Gary G. Yen, *Fellow, IEEE*, and Biruk G. Tessema

***Abstract*—This paper proposes a constraint handling technique for multiobjective evolutionary algorithms based on an adaptive penalty function and a distance measure. These two functions vary dependent upon the objective function value and the sum of constraint violations of an individual. Through this design, the objective space is modified to account for the performance and constraint violation of each individual. The modified objective functions are used in the nondominance sorting to facilitate the search of optimal solutions not only in the feasible space but also in the infeasible regions. The search in the infeasible space is designed to exploit those individuals with better objective values and lower constraint violations. The number of feasible individuals in the population is used to guide the search process either toward finding more feasible solutions or favor in search for optimal solutions. The proposed method is simple to implement and does not need any parameter tuning. The constraint handling technique is tested on several constrained multiobjective optimization problems and has shown superior results compared to some chosen state-of-the-art designs.**

***Index Terms*—Constraint handling, evolutionary multiobjective optimization, genetic algorithm.**

I. Introduction

EVOLUTIONARY algorithms (EAs) have been successfully applied to solve optimization problems in the fields of science and engineering. In addition to solving unconstrained optimization problems, researchers have been able to tailor constraint handling techniques into these algorithms. The challenges in constrained optimization problems arise from the various limits on the decision variables, the constraints involved, the interference among constraints, and the interrelationship between the constraints and the objective functions [1]. In the meantime, researchers were also developing evolutionary approaches for solving multiobjective optimization problems (MOPs) [2], [3]. A historical review of evolutionary multiobjective optimization is presented in [4]. These multiobjective evolutionary algorithms (MOEAs) are capable of simultaneously optimizing a set of competing objectives. Nevertheless, little research was conducted in the area of constrained multiobjective optimization [5]. Such problems involve multiple conflicting objectives that are subject to various equality and inequality constraints.

Manuscript received September 05, 2008; revised February 18, 2008; July 05, 2008; and August 15, 2008. First published March 10, 2009; current version published date June 10, 2009.

The authors are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: gyen@okstate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2008.2009032

A constrained multiobjective optimization problem (CMOP) can be mathematically formulated as

$$\begin{aligned} &\frac{\text{Minimize}}{\text{Maximize}} \\ &\qquad f_i(x) = f_i(x_1, x_2, \ldots, x_n), \quad i = 1, \ldots, k \\ &\text{subject to } g_j(x) = g_j(x_1, x_2, \ldots, x_n) < 0, \; j = 1, \ldots, q \\ &\qquad h_j(x) = h_j(x_1, x_2, \ldots, x_n) = 0, \; j = q+1, \ldots, m \\ &\qquad x_j^{\min} \le x_j \le x_j^{\max}, \; j = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

There are k objective functions that are required to be simultaneously optimized. Each objective function $f_i(x)$ is defined on the search space $S \subseteq \Re^n$. Usually, the search space is an n-dimensional hyperbox in $\Re^n$. Each dimension of the search space is bounded by its upper $(x_j^{\max})$ and lower $(x_j^{\min})$ limits.

$g_j(x)$ is the j^{th}-inequality constraint, and $h_j(x)$ is the j^{th}-equality constraint. There are a total of m constraints, q inequality and $m-q$ equality, which are required to be satisfied by the optimum solution. The presence of equality and inequality constraints will restrict the search space to a feasible region $F \subseteq S$, where a usable solution can be found.

This paper extends the single-objective constrained optimization algorithm proposed by Tessema and Yen [1] to CMOPs. The proposed algorithm basically modifies the objective function of an individual using its distance measure and penalty value. These modified objective function values are ranked through the nondominance sorting of the multiobjective optimization. Distance measures are found for each dimension of the objective space by incorporating the effect of an individual's constraint violation into its objective function. The penalty function, on the other hand, introduces additional penalty for infeasible individuals based on their objective values and constraint violations. The balance between two components, one based on objective function and the other on constraint violation, is controlled by the number of feasible individuals currently present in the population. If few feasible individuals are present, then those infeasible individuals with higher constraint violations are penalized more than those with lower constraint violations. On the other hand, if a sufficient number of feasible individuals exists, then those infeasible individuals with worse objective values are penalized more than those with better objective values. However, if the number of feasible individuals is in the middle of the two extremes, then the individual with lower constraint violation and better objective function is less penalized. The two components of the penalty function allow the algorithm to switch between feasibility and optimality at anytime during the evolutionary process. Furthermore, since priority is initially given to finding feasible individuals before searching for optimal solutions, the algorithm is capable of finding feasible solutions when the feasible space is very small compared to the search space.

This paper is structured as follows. Section II provides a brief overview of the various evolutionary approaches developed for CMOPs. Then, in Section III, the proposed CMOP evolutionary algorithm is presented and analyzed in detail. Next, in Section IV, we discuss various CMOP test problems and use them to evaluate the proposed algorithm as opposed to two chosen state-of-the-art designs in literature. Finally, we present the results of the experiments and conclude with a summary of this paper and ideas for future work.

II. Literature Survey

This section presents a brief review of evolutionary approaches developed for CMOPs. Over the last decade, several MOEAs have been developed to solve multiobjective optimization problems. The earlier MOEAs are non-elitism-based methods that assign fitness to population members based on nondominated sorting. In addition, they exploit different techniques to preserve diversity among solutions of the same nondominated front. Of these types, the Multiobjective Genetic Algorithm (MOGA) [6] by Fonseca and Fleming and the Nondominated Sorting Genetic Algorithm (NSGA) [7] by Srinivas and Deb are very popular [4]. An improved version of NSGA, called NSGA-II, was later proposed by Deb *et al.* [8] for better performance.

More recently, elitism-based algorithms have been suggested to enhance the convergence properties of MOEAs. The Pareto Archived Evolution Strategy (PAES) [9] by Knowles and Corne uses a $(1+1)$ evolution strategy with a historical archive that records all the nondominated solutions found until the current generation. It also designs a novel approach to maintain diversity which consists of a crowding procedure that divides objective space in a recursive manner into several grids. This procedure is adaptive and has lower computational complexity than the traditional niching-based approaches. Zitzler and Thiele introduce the Strength Pareto Evolutionary Algorithm (SPEA) [10] that uses an external archive to preserve nondominated solutions. In each generation, the nondominated solutions in the external set are given a strength value that is proportional to the number of individuals they dominate. Fitness of individuals in the main population are computed according to the strengths of all external nondominated solutions that dominate it. In addition, a clustering technique is used to preserve diversity. Today, many advanced versions of MOEAs are made available in continued pursuit of better performance and a vast majority of these literatures do not involve constraints [5].

On the other hand, constraint handling for single objective optimization problems has also been actively researched over the past two decades [11], [12]. Penalty functions are the simplest and most commonly used methods for handling constraints using EAs. In death penalty function methods such as [13], individuals that violate any one of the constraints are rejected and no information is extracted from infeasible individuals. If the added penalties do not depend on the current generation number and remain constant during the entire evolutionary process, then the penalty function is called static penalty function. In static penalty function methods [14], the penalties are the weighted sum of the constraint violations. If, alternatively, the current generation number is considered in determining the penalties, then the method is called dynamic penalty function method [15]. In adaptive penalty function methods [16], [17], information gathered from the search process will be used to control the amount of penalty added to infeasible individuals.

In [15] and [18], methods based on preference of feasible solutions over infeasible solutions are employed. In these types of techniques, feasible solutions are always considered better than infeasible ones. Therefore, when population fitness ranking is performed, feasible individuals will come first followed by infeasible individuals with low constraint violation. In [19] and [20], Runarsson and Yao introduce the stochastic ranking method to achieve a balance between objective and penalty functions stochastically. A probability factor is used to determine whether the objective function value or the constraint violation value determines the rank of each individual. In [21] and [22], similar algorithms are proposed where constraint violation and objective function are optimized separately.

More recently, multiobjective optimization techniques have been used to solve constrained optimization problems. In [23], a multiobjective optimization technique that uses population-based algorithm generator and infeasible solutions archiving and replacement mechanism is introduced. In [24], a two-phase algorithm that is based on multiobjective optimization technique is proposed. In the first phase of the algorithm, the objective function is completely disregarded and the constraint optimization problem is treated as a constraint satisfaction problem. In the second phase, both constraint satisfaction and objective optimization are treated as a bi-objective optimization problem. An algorithm that combines penalty function approach and multiobjective optimization technique is also suggested in [25]. The algorithm has a similar structure as the penalty-based approach but borrows the ranking scheme from multiobjective optimization techniques.

Although multiobjective optimization and constraint handling have received a lot of attention individually, very little effort has been devoted in solving CMOPs. Coello Coello and Christiansen [26] propose a naïve approach to solve CMOPs by ignoring any solution that violates any of the assigned constraints. This method is easy to implement but it may experience difficulty in finding feasible solutions. This is due to the reason that rejecting infeasible individuals in the vicinity of feasible solutions will impair the search capability which eventually affects the algorithm's ability to find feasible solutions. This can be observed when a small feasible region is surrounded by infeasible solutions.

In [27], Binh and Korn propose the Multiobjective Evolution Strategy (MOBES), which takes into account the objective function vector as well as the degree of constraint violation of infeasible solutions in order to evaluate their fitness. Infeasible individuals will be divided into different classes according to their "nearness" to the feasible region, and ranking will be performed based on the class. In addition, a mechanism to maintain a feasible Pareto optimal set is employed.

In [8], Deb *et al.* propose a constrained multiobjective algorithm based on constrained dominance of individuals. According to their algorithm, a solution i is said to constrained-dominate a solution j if 1) i is feasible, while j is infeasible; 2) both are infeasible and i has less constraint violation; or 3) both are feasible and i dominates j. Feasible solutions constrained-dominate all infeasible solutions. However, when two feasible individuals are compared, the usual dominance relationship is used. The level of constraint violation is used to compare two infeasible individuals.

In [28], Jimenez *et al.* propose the Evolutionary Algorithm of Nondominated Sorting with Radial Slots (ENORA), which employs the min–max formulation for constraint handling. Feasible individuals evolve toward optimality, while infeasible individuals evolve toward feasibility. In addition, a diversity technique based on partitioning of the search space in a set of radial slots along which the successive populations generated by the algorithm are positioned is introduced.

In [29], Ray *et al.* suggest using three different nondominated rankings of the population. In this algorithm, which will be referred to as Ray–Tai–Seow algorithm in Section IV, the first ranking is performed using the objective function values; the second is performed using different constraints; and the last ranking is based on the combination of all objective functions and constraints. Depending on these rankings, the algorithm performs according to the predefined rules. In [30], Chafekar *et al.* propose two novel approaches for solving constrained multiobjective optimization problems. One method, called Objective Exchange Genetic Algorithm of Design Optimization (OEGADO), runs several GAs concurrently with each GA optimizing one objective and exchanging information about its objective with others. The other method, called Objective Switching Genetic Algorithm for Design Optimization (OSGADO), runs each objective sequentially with a common population for all objectives.

In [31], Young proposes a constrained multiobjective evolutionary algorithm called Blended Space EA (BSEA). The algorithm checks dominance by using a rank obtained by blending an individual's rank in objective space with its rank in constraint space. A similar approach is proposed by Angantyr *et al.* [32] that uses the weighted average rank of the ranks in the constraint and objective space. Although their algorithm was examined only for testing problems with one objective function and several constraints, a simple adjustment in their formulation will provide a constrained multiobjective optimization tool.

Fonseca and Fleming [33] propose a unified approach for multiobjective optimization and multiple constraint handling. Their algorithm handles constraints by assigning high priority to constraints and low priority to objective functions, which allows search of feasible solutions followed by search of optimal solutions.

Harada *et al.* [34] propose Pareto Descent Repair (PDR) operator that searches for feasible solutions out of infeasible individuals in the constraint function space. This operator involves gradients that are usually unavailable in functions justified to be optimized using EAs.

In light of superior performance achieved in [1] for the single objective constraint optimization, a similar idea is extended in this paper into the uses of multiobjective constraint optimization. In the next section, we introduce the proposed Constrained Multiobjective Evolutionary Algorithm (CMOEA).

III. Proposed Algorithm

The proposed algorithm extends the single-objective constrained evolutionary algorithm proposed by Tessema and Yen [1] into a multiobjective framework. The major difference in various constraint handling techniques used in multiobjective optimization arises from the variations in the involvement of infeasible individuals in the evolutionary process. The main purpose of involving infeasible individuals in the search process is to exploit the information they carry. Since EAs are stochastic search techniques, discarding infeasible individuals might lead to the EA being stuck in local optima, especially in problems with disjoint search space. In addition, in some highly constrained optimization problems, finding a single feasible individual by itself might be a daunting challenge when the algorithm must be able to extract information from the previous infeasible individuals.

The proposed algorithm uses modified objective function values for checking dominance in the population. The modification is based on the constraint violation of the individual and its objective performance. The modified objective value has two components: distance measure and adaptive penalty. The two components are discussed next in detail.

A. Distance Values

Distance measure is found for each dimension of the objective space by including the effect of an individual's constraint violation into its objective function. The major steps in calculating the distance measure starts with obtaining the minimum and maximum values of each objective function in the population

$$f^i_{\min} = \min_x f_i(x) \tag{2a}$$

and

$$f^i_{\max} = \max_x f_i(x). \tag{2b}$$

Using these values, normalize each objective function i for every individual x

$$\tilde{f}_i(x) = \frac{f_i(x) - f^i_{\min}}{f^i_{\max} - f^i_{\min}} \tag{3}$$

where $\tilde{f}_i(x)$ is the normalized i^{th}-objective value of individual $x, i = 1, \ldots, k$.

Constraint violation, $v(x)$, of individual x is then calculated as the summation of the normalized violations of each constraint divided by the total number of constraints

$$v(x) = \frac{1}{m} \sum_{j=1}^{m} \frac{c_j(x)}{c^j_{\max}} \tag{4}$$

where

$$c_j(x) = \begin{cases} \max(0, g_j(x)) & j = 1, \ldots, q \\ \max(0, |h_j(x)| - \delta) & j = q+1, \ldots, m \end{cases} \tag{5a}$$

$$c^j_{\max} = \max_x c_j(x) \tag{5b}$$

δ is the tolerance value for equality constraints (usually 0.001 or 0.0001). q is the number of inequality constraints, and $m-q$ is the number of equality constraints. If the constraint violation $c_j(x)$ is greater than zero, then the individual x violates the j^{th}-constraint. On the other hand, if the constraint violation $c_j(x)$ is equal to zero, then the individual x satisfies the j^{th}-constraint and the constraint violation $c_j(x)$ is set to zero.

Then the "distance" value of individual x in each objective function dimension i is formulated as follows:

$$d_i(x) = \begin{cases} v(x), & \text{if } r_f = 0 \\ \sqrt{\tilde{f}_i(x)^2 + v(x)^2}, & \text{otherwise} \end{cases} \tag{6}$$

where

$$r_f = \frac{\text{number of feasible individuals in current population}}{\text{population size}}. \tag{7}$$

Interested readers are referred to the example in [1] for how to compute the distance value. The pseudocode for calculating the distance value is given in Fig. 4. From (6), we observe that if there is no feasible individual in the current population, then the distance values are equal to the constraint violation of the individual. In this case, according to the distance values, an infeasible individual with smaller constraint violation will dominate another infeasible individual with higher constraint violation regardless of their objective function values. This is the best way to compare infeasible individuals in the absence of feasible individuals [18], [21] since it gives priority to finding feasible individuals over finding optimal solutions. On the other hand, if there is more than one feasible solution in the population, then the distance values will have the properties summarized below.

1) For a feasible individual x, the distance value in a given objective function dimension i is equal to $\tilde{f}_i(x)$. Hence, those feasible individuals with smaller objective function values will have smaller distance values in that given dimension.
2) For infeasible individuals, the distance value has two components: the objective function value and the constraint violation. Hence, individuals closer to the origin in the $\tilde{f}_i(x) - v(x)$ space would have lower distance value in that objective function dimension than those farther away from the origin.
3) If we compare the distance values of infeasible and feasible individuals, then either one may have a smaller value. However, if the two individuals have similar objective function values, then the feasible individual will have a smaller distance value in the corresponding objective function dimension.

B. Two Penalties

In addition to the distance measure, two penalty functions are added to the fitness value of infeasible individuals. These functions penalize infeasible individuals based on their corresponding objective value and constraint violation. The first penalty function is based on the objective functions, and the second is based on the constraint violation. The balance between the two components is controlled by the number of feasible individuals currently present in the population.

These penalties have two major purposes.

1) To further reduce the fitness of infeasible individuals as the penalty imposed by the distance formulation alone is small.
2) To identify the best infeasible individuals in the population by adding different amount of penalty to each infeasible individual's fitness.

The two penalties are formulated for individual x in the i^{th}-objective function dimension as follows:

$$p_i(x) = (1 - r_f)X_i(x) + r_f Y_i(x) \tag{8}$$

where

$$X_i(x) = \begin{cases} 0, & \text{if } r_f = 0 \\ v(x), & \text{otherwise} \end{cases} \tag{9a}$$

and

$$Y_i(x) = \begin{cases} 0, & \text{if } x \text{ is a feasible individual} \\ \tilde{f}_i(x), & \text{if } x \text{ is an infeasible individual} \end{cases}. \tag{9b}$$

From the penalty function definition in (8) and (9), we observe that if the feasibility ratio of the population is small (but not zero), then the first penalty $(X_i(x))$ will have more impact than the second penalty $(Y_i(x))$. The first penalty is formulated to have large value for individuals with large amount of constraint violation. Hence, in the case when there are few feasible individuals present in the population (r_f is small), infeasible individuals with higher constraint violation will be more penalized than those with lower constraint violation. On the other hand, if there are many feasible solutions in the population (r_f is large), the second penalty will have more effect than the first. In this case, infeasible individuals with larger objective function value will be more penalized than infeasible individuals with smaller objective function value. Additionally, if there are no feasible individuals in the population ($r_f = 0$), both penalties will be zero.

The two components of the penalty function allow the algorithm to switch between finding more feasible solutions and finding better solutions at anytime during the evolutionary process. Furthermore, because priority is initially given to the search for feasible individuals, the algorithm is capable of finding feasible solutions in cases where the feasible space is small or disjoint compared to the search space. This argument will be demonstrated through testing functions in Section IV. The pseudocode for calculating the penalty value is given in Fig. 5.

C. Final Modified Objective Value Formulation

The final modified objective value of individual x, using which nondominance sorting is performed, is formulated as the sum of the distance measure and penalty function in the i^{th}-objective function dimension

$$F_i(x) = d_i(x) + p_i(x). \tag{10}$$

This modified objective value formulation is flexible and will allow us to utilize infeasible individuals efficiently and effectively. Most constraint optimization algorithms in literature are

TABLE I
SUMMARY FOR THE BASIC CHARACTERSTICS OF THE BENCHMARK TEST PROBLEMS USED IN THIS PAPER

Function name	Objective functions	Decision dimensions	Feasibility ratio (ρ)	Constraints				
				Inequality	Equality	Linear	Non-linear	Active
BNH	2	2	93.61%	2	0	0	2	0
SRN	2	2	16.18%	2	0	1	1	0
OSY	2	6	3.25%	6	0	4	2	3
TNK	2	2	5.09%	2	0	0	2	1
CTP1	2	2	99.58%	2	0	0	2	1
CTP2	2	2	78.65%	1	0	0	1	1
CTP3	2	2	76.85%	1	0	0	1	1
CTP4	2	2	58.17%	1	0	0	1	1
CTP5	2	2	77.54%	1	0	0	1	1
CTP6	2	2	0.40%	1	0	0	1	1
CTP7	2	2	36.68%	1	0	0	1	0
CTP8	2	2	17.83%	2	0	0	2	1
CONSTR	2	2	52.52%	2	0	2	0	1
Welded Beam	2	4	18.67%	4	0	1	3	0

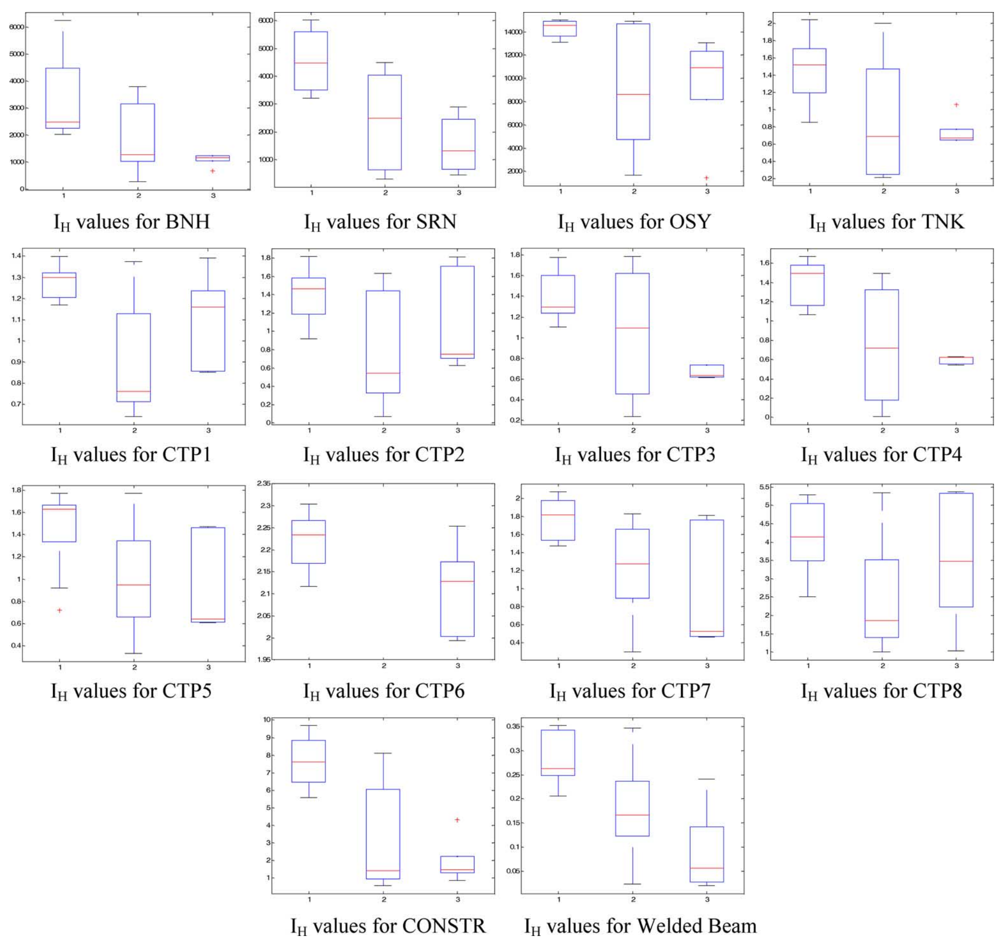


Fig. 1. Box plot of hypervolume indicator (I_H values) for all test functions by algorithms 1–3 represented (in order): the Proposed algorithm, NSGA-II, and Ray–Tai–Seow's.

"rigid" in a sense that they always prefer certain types of infeasible individuals throughout the entire evolutionary process. For example, they might always give priority to those individuals with small constraint violation only or those individuals with low objective value only. According to our new fitness formulation, the infeasible individuals that are considered valuable are not always similar. Here are some of the interesting properties of this modified objective value formulation.

1) If there is no feasible individual in the current population, each $d_i(x)$ will be equal to the constraint violation $(v(x))$, and each $p_i(x)$ term will be zero. In this case, the objective values of the individuals will be totally disregarded, and

TABLE II

THE DISTRIBUTION OF $\mathrm{I_H}$ VALUES TESTED USING MANN–WHITNEY RANK-SUM TEST [40]. THE TABLE PRESENTS THE p-VALUES WITH RESPECT TO THE ALTERNATIVE HYPOTHESIS (I.E. $p\text{-value} < \alpha = 0.05$) FOR EACH PAIR OF THE PROPOSED ALGORITHM AND A SELECTED CMOEA. THE DISTRIBUTION OF THE PROPOSED ALGORITHM HAS SIGNIFICANT DIFFERENCES THAN THOSE SELECTED CMOEA UNLESS STATED

Test Functions	BNH	SRN	OSY	TNK	CTP1	CTP2	CTP3
$\mathrm{I_H}$ (Proposed, NSGA-II)	0.0315	0.0244	0.1081 > 0.05 no difference	0.0315	0.0078	0.04	0.2224 > 0.05 no difference
$\mathrm{I_H}$ (Proposed, Ray-Tai-Seow)	4.11E-05	4.11E-05	4.11E-05	1.65E-04	0.0476	0.3284 > 0.05 no difference	4.11E-05
Test Functions	**CTP4**	**CTP5**	**CTP6**	**CTP7**	**CTP8**	**CONSTR**	**Welded Beam**
$\mathrm{I_H}$ (Proposed, NSGA-II)	0.0051	0.0315	No feasible soln	0.04	0.0625 > 0.05 no difference	0.0056	0.0078
$\mathrm{I_H}$ (Proposed, Ray-Tai-Seow)	4.11E-05	0.0027	0.0228	0.0175	0.7785 > 0.05 no difference	4.11E-05	5.35E-04

TABLE III

THE DISTRIBUTION OF $\mathrm{I}\varepsilon+$ VALUES TESTED USING MANN–WHITNEY RANK-SUM TEST [40]. THE TABLE PRESENTS THE p-VALUES WITH RESPECT TO THE ALTERNATIVE HYPOTHESIS (I.E., $p\text{-value} < \alpha = 0.05$) FOR EACH PAIR OF THE PROPOSED AND A SELECTED CMOEA. THE DISTRIBUTION OF THE PROPOSED ALGORITHM HAS SIGNIFICANT DIFFERENCES THAN THOSE SELECTED CMOEA UNLESS STATED

Test Functions	BNH	SRN	OSY	TNK	CTP1	CTP2	CTP3
Iε+ (Proposed, NSGA-II)	4.11E-05	4.11E-05	4.11E-05	2.00E-03	0.1359 > 0.05 no difference	4.11E-05	4.11E-05
I ε+ (Proposed, Ray-Tai-Seow)	8.23E-05	4.11E-05	4.11E-05	1.42E-02	0.3734 > 0.05 no difference	4.11E-05	1.65E-04
Test Functions	**CTP4**	**CTP5**	**CTP6**	**CTP7**	**CTP8**	**CONSTR**	**Welded Beam**
Iε+ (Proposed [illegible]	0.4755 > 0.05 [illegible]		no feasible [illegible]		0.5457 > 0.05 [illegible]		

all individuals will be compared based on their constraint violation only. This will help us find feasible individuals before looking for optimal solutions.

2) If there are feasible individuals in the population, then individuals with both low objective function values and low constraint violation values will dominate individuals with high objective function values or high constraint violation or both.
3) If two individuals have equal or very close distance values, then the penalty term $(p_i(x))$ determines the dominant individual. According to our penalty formulation, if the feasibility ratio (r_f) in the population is small, then the individual closer to the feasible space will be dominant. On the other hand, the individual with smaller objective function values will be dominant. Otherwise, the two individuals will be nondominant solutions.
4) If there is no infeasible individual in the population ($r_f = 1$), then individuals will be compared based on their objective function values alone.

After the computation of the modified objective values, the standard features of NSGA-II, such as nondominant ranking and diversity through crowding distances, will be used based on these modified values. During the archiving process, which stores the best individuals from the current population, the best feasible individuals are given priority for archiving over infeasible individuals, as the goal of constrained multiobjective optimization is eventually to find feasible optimal solutions. The general pseudocode for the proposed algorithm is given in Fig. 6. The proposed constraint handling technique is very generic, involving only the modification of fitness function through adaptive penalty measure.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Experimental Setup

The proposed algorithm is tested on several constrained multiobjective benchmark problems available from literature. The simulations are conducted with a population size of 100, crossover rate of 0.8, mutation rate of 0.2, and maximum generation number of 100 for all implementations. In addition, we use SBX crossover and mutation. Tournament selection is adopted in a recombination and replacement scheme. These design parameters are chosen to be consistent with what were used in [35]. Fourteen benchmark problems have been selected to evaluate the performance of the proposed algorithm. These problems have been used in [29] and [35]. These problems

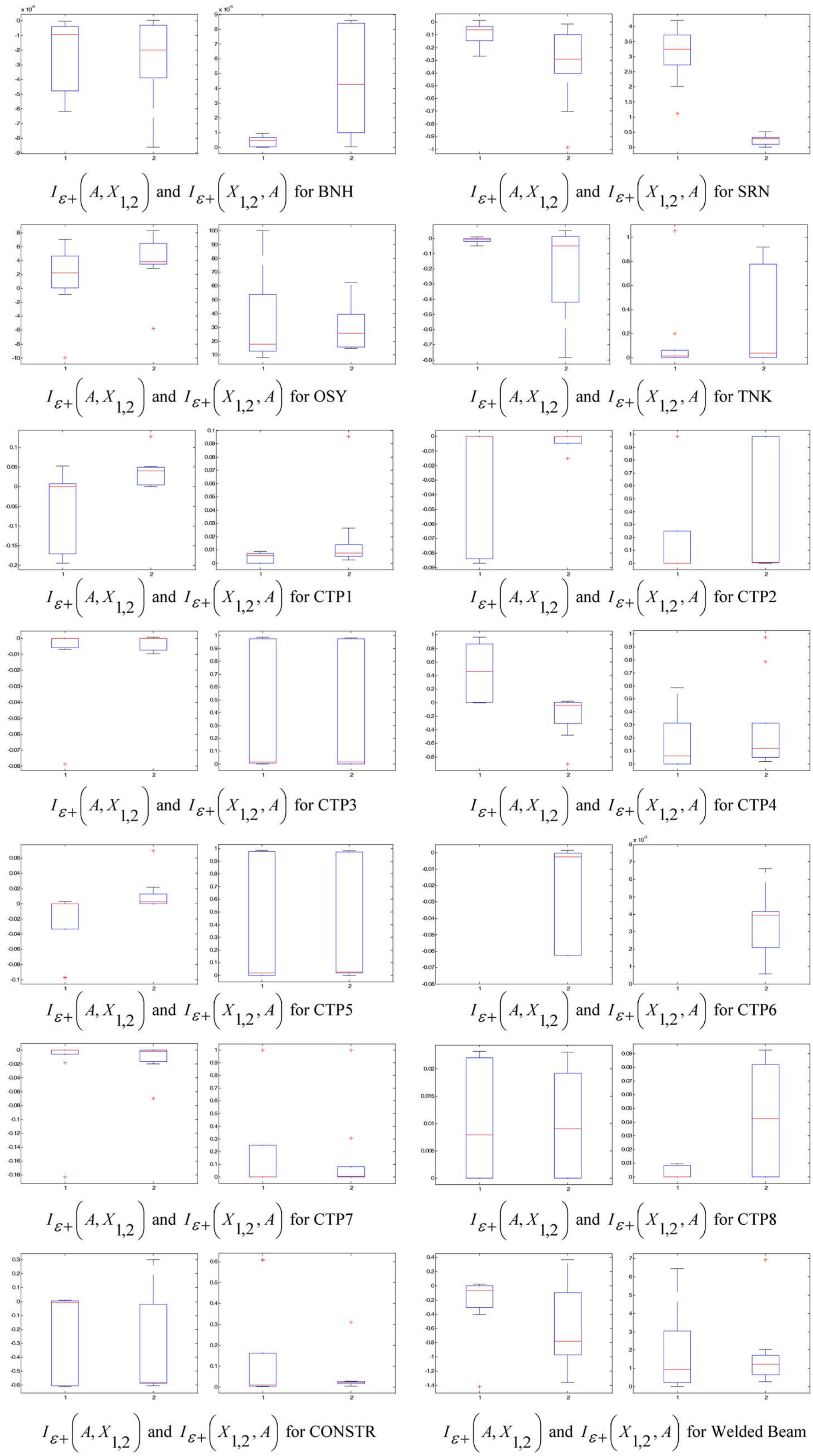


Fig. 2. Box plots of additive epsilon indicator ($I\varepsilon+$ values) ("A" corresponds to the proposed algorithm, while '$X_{1,2}$' refers to NSGA-II and Ray–Tai–Seow's, respectively).

are all minimization problems and are denoted as BNH [27], SRN [7], [36], OSY [37], TNK [38], CTP1 [35], CTP2 [35], CTP3 [35], CTP4 [35], CTP5 [35], CTP6 [35], CT7 [35], CTP8 [35], CONSTR [35], and Welded Beam Problem [29]. The problem characteristics for these test problems are summarized in Table I. The welded beam problem represents a real world optimization problem [29]. Some of the test problems have continuous Pareto-fronts (BNH, SRN, OSY, CTP1, CTP6, CONSTR and Welded Beam), while the remaining problems have disjoint Pareto-fronts (TNK, CTP2-CTP5, CTP7, CTP8). It presents a great challenge to the proposed algorithm to locate disjoint Pareto fronts with alternating feasible and infeasible regions. Among the test problems, the CTP8 benchmark problem possesses multiple small and disjoint feasible regions that impose greater difficulty in locating the optimal solutions. As can be seen from the feasibility ratio listed in Table I, the feasibility region estimated for this problem is a fraction of the search space. The feasibility ratio is determined experimentally by calculating the percentage of feasible solutions among 1 000 000 randomly generated individuals [24]. Please note that there exists no benchmark function previously used in EA literatures involving *equality constraints* for constrained multiobjective optimization algorithms, to our best knowledge.

Each benchmark problem is run 50 times as commonly adopted in literature [35] and the performance metrics are measured statistically. Both quantitative and qualitative comparisons are made to validate the proposed algorithm. For qualitative comparison, the plots of final nondominated fronts that were obtained from the same initial population are presented. The quantitative comparison is performed using hypervolume indicator and additive epsilon indicator. These two Pareto compliant performance metrics are able to measure the performance of algorithms with respect to their dominance relations and diversity preservation. A detailed discussion about these measures can be found in [39] and [40]. The quantitative comparisons are illustrated by statistical box plots, and a Mann–Whitney rank-sum test is implemented to evaluate whether the difference in performance between two independent samples is significant [40].

B. Comparative Study

The performance metric for hypervolume indicator (I_H value) is computed for each CMOEA over 50 independent runs. Fig. 1 presents the box plots of I_H indicator found in all CMOEAs, in which 1 is denoted by the proposed algorithm, 2 as the NSGA-II and 3 as the Ray-Tai-Seow's. Higher I_H value indicates the ability of the algorithm to dominate a larger region in the objective space. The figure shows that the proposed algorithm has the highest I_H values for the test functions BNH, SRN, TNK, CTP1, CTP4, CTP5, CTP6, CTP7, CONSTR, and Welded Beam. The proposed algorithm and NSGA-II showed comparable I_H values for test problems OSY and CTP3. Similarly, Ray–Tai–Seow's algorithm showed comparable I_H values for CTP2 test problem. All algorithms performed well for CTP8 test problem. NSGA-II showed a higher I_H value than Ray–Tai–Seow for test functions SRN, CTP3, CTP5, CTP7, and welded Beam. On the contrary, Ray–Tai–Seow's showed better I_H value than NSGA-II for test functions OSY, CTP1, CTP2, and CTP8. As can be seen later from Fig. 3, NSGA-II was not able to find a feasible solution for CTP6 test problem due to the extent of constraints imposed.

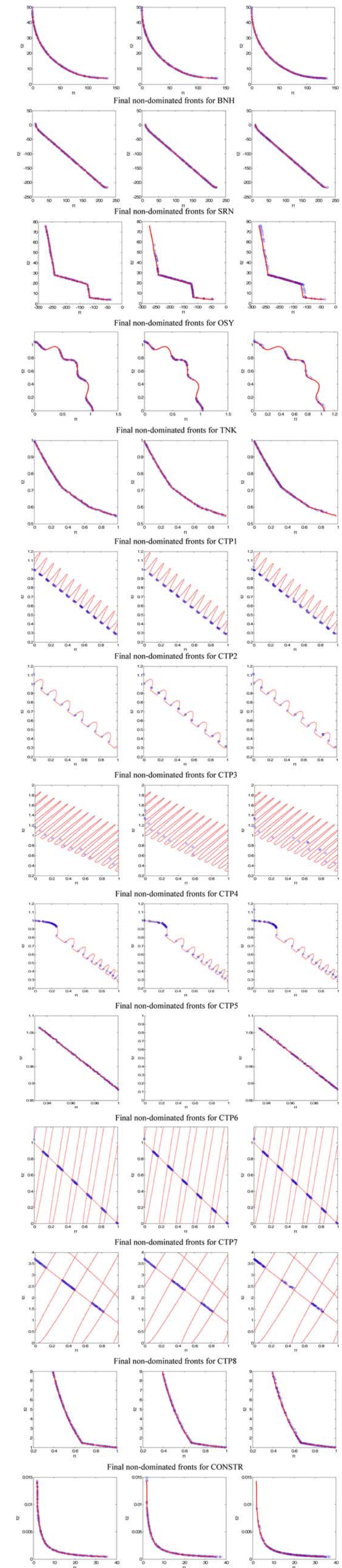


Fig. 3. Final nondominated fronts plots for all test functions by the proposed algorithm (on the left), NSGA-II (in the middle), and Ray–Tai–Seow's (on the right).

In some of the problems shown in Fig. 1, it is hard to determine whether the proposed algorithm is significantly better than

```
Begin
    If r_f = 0 then
        For i = 1 to number_of_objectives Do
            For k = 1 to Population_Size Do
                d_i(x_k) ← v(x_k)
            End For
        End For
    Else
        For i = 1 to number_of_objectives Do
            For k = 1 to Population_Size Do
                f̃_i(x_k) ← (f_i(x_k) − f^i_min) / (f^i_max − f^i_min)
                d_i(x_k) ← sqrt( f̃_i(x_k)^2 + v(x_k)^2 )
            End For
        End For
    End If
End
```

Fig. 4 Psuedocode for finding distance value.

```
Begin
    For i = 1 to number_of_objectives Do
        For k = 1 to Population_Size Do
            If r_f = 0 then
                X_i(x_k) ← 0
            Else
                X_i(x_k) ← v(x_k)
            End If

            If v(x_k) = 0 then
                Y_i(x_k) ← 0
            Else
                Y_i(x_k) ← f̃_i(x_k)
            End If
            p_i(x_k) ← (1 − r_f) X_i(x_k) + r_f Y_i(x_k)
        End For
    End For
End
```

Fig. 5 Psuedocode for finding penalty value.

the other CMOEAs since they attain close $\mathrm{I_H}$ values. Hence, the Mann–Whitney rank-sum test is used to examine the distribution of the $\mathrm{I_H}$ values. The tested results are presented in Table II, and they indicate that the proposed algorithm's performance has a significant advantage compared to the distribution in NSGA-II and Ray–Tai–Seow's in most test functions except OSY, CTP2, CTP3, and CTP8. In addition, Fig. 1 shows that the standard deviations for the proposed algorithm are consistently lower, which indicates the proposed algorithm is more reliable in producing better solutions than those selected CMOEAs, i.e., NSGA-II and Ray–Tai–Seow algorithms.

Fig. 2 illustrates the results of additive epsilon indicator using statistical box plots. This indicator, recommended in [39], is capable of detecting whether a nondominated set is better than another. There are two box plots for each test problem, i.e., $I_{\varepsilon+}(A, X_{1,2})$ and $I_{\varepsilon+}(X_{1,2}, A)$, in which algorithm A is referred to as the proposed design, while algorithms 1 and 2 represent NSGA-II and Ray–Tai–Seow's, respectively. It seems that the proposed algorithm performs relatively better with respect to dominance relation than most of the CMOEAs for all functions except OSY, CTP2, CTP4, CTP5, and CTP8. For example, Fig. 2 shows that the proposed algorithm strictly dominates NSGA-II on Welded Beam problem because $I_{\varepsilon+}(A, X_1) \leq 0$ and $I_{\varepsilon+}(X_1, A) > 0$. On the other hand, the box plot on CTP8 in Fig. 2 may indicate that the proposed algorithm does not strictly dominate NSGA-II because $I_{\varepsilon+}(A, X_1) > 0$ and $I_{\varepsilon+}(X_1, A) > 0$. In summary, NSGA-II and the proposed algorithm showed no difference for CTP4 test function; Ray–Tai–Seow's and the proposed algorithm showed comparable results for CTP2 and CTP5 test functions; and finally the proposed algorithm seems to perform as well as NSGA-II and Ray–Tai–Seow's for functions OSY and CTP8. For the rest of the test functions, the proposed algorithm showed better performance compared to the other CMOEAs. Moreover, we can observe that the proposed algorithm has relatively lower standard deviations, which are consistent with those shown in Fig. 1.

For further analysis, the distributions of $I_{\varepsilon+}$ values are analyzed via the Mann–Whitney rank-sum test, which are presented in Table III. In general, results in Table III and Fig. 2 confirm that the proposed algorithm is significantly better than most or even all of the CMOEAs on all benchmark test problems in terms of the chosen performance metrics. For qualitative comparison, all the CMOEAs are initialized from the same population and the resulting Pareto fronts are plotted from a single run. The resulting final nondominated fronts for all of the 14 test functions are illustrated in Fig. 3. For each test function, three plots are presented: the one on the left is from the proposed algorithm, the one in the middle is from NSGA-II, while the one on the right is from Ray–Tai–Seow's. The figures show the proposed algorithm is able to find the well-extended, well-spread, and near-optimal Pareto fronts for all test functions.

Please note the true Pareto fronts for all 14 benchmark functions are marked in solid lines in Fig. 3, while the resulted nondominated individuals from each of the CMOEAs are shown in small circles in these plots. The plots of the obtained nondominated fronts for the constrained multiobjective test problems meet our expectation. Not only are we able to find feasible individuals, but we are also able to find better-fit individuals that are on or very close to the true Pareto front. In addition, the proposed algorithm provides a more evenly distributed and well extended Pareto front based on visual inspection of Fig. 3 and the results from Hypervolume indicators shown in Fig. 1. The success of the proposed algorithm can be attributed to the exploitation of the evolutionary information contained in infeasible individuals in addition to that contained in feasible individuals. This is done by allowing infeasible individuals to participate in the evolutionary process. The constraint handling normally used in NSGA-II [8] compares infeasible individuals solely based on their constraint violation. This way of nondominance ranking ignores how well each individual performed in the objective space and may result in the inefficient use of some evolutionary materials. The proposed algorithm, on the other hand, uses a combined measure of constraint violation and objective performance to arrive at the fitness of individuals that

```
/* Proposed Constrained Multi-Objective Evolutionary Algorithm */
Begin
    Initialize N solutions
    Evaluate all individuals
    /* Constraint satisfaction */
    Do While ("no feasible solution is found" or "maximum generation is reached")
        1) Give fitness to individuals based on their sum of constraint violations.
        2) Rank individuals based on fitness in 1.
        3) Selection, Recombination, Mutation and Replacement.
        4) Archive if any feasible solutions is found.
    End Do
    /* Feasible solutions have been found */
    /* Constraint satisfaction and objective optimization */
    Do While ("maximum generation is reached")
        5) Calculate modified objective function values using distance measures and penalty
           functions for all individuals.
        6) Pareto sort individuals according to their modified objective function values.
        7) Give fitness to individuals according to Pareto ranking and crowding distance.
        8) Use tournament selection to select N parents.
        9) Generate N offspring solutions.
        10) Calculate fitness of offspring solutions.
        11) Update archive. If a feasible offspring dominates a solution in the archive, then it will
            replace that solution.
        12) Trim the main population to N individuals based on the fitness of the individuals.
    End Do
End
```

Fig. 6 Psuedocode for proposed algorithm.

will govern the evolutionary process. The number of feasible individuals available in the current population is used to control the relative emphasis given to either constraint violation or objective performance in the final fitness calculation. As can be observed from the test results of the proposed algorithm, this way of fitness formulation provides better solutions compared to other constrained multiobjective evolutionary algorithms.

V. Conclusion and Discussions

In this paper, we propose an adaptive constraint handling technique for solving CMOPs. Besides the search for optimal solutions in the feasible region, the algorithm also exploits the information hidden in infeasible individuals with better objectives and lower constraint violation. This is achieved by using the modified objective values in the nondominance ranking of the multiobjective evolutionary algorithm. The modified objective values are composed of distance measures and penalty functions. These values are associated with how well an individual performs and how much it violates the constraints. The number of feasible individuals in the population adaptively controls the emphasis given to objective values or constraint violation in the modified objective function formulation. If there is no feasible individual in the population, the algorithm uses the constraint violations as the primary means to rank the individuals. Involving infeasible individuals in the evolutionary process helps the algorithm to find additional feasible individuals, even in cases where the feasible space is very small and disjoint. Furthermore, since there is no parameter tuning, this makes the algorithm easy to implement. Moreover, the additional evaluations are simple arithmetic operations and do not impose any significant increase in the computational cost.

The proposed constraint handling technique is implemented on NSGA-II simply due to its popularity as an MOEA. The proposed constraint handling technique can be easily extended to other MOEAs. The performance of the algorithm was tested on 14 constrained multiobjective test problems. From the simulation results, it is observed that the algorithm is capable of finding better-fit feasible solutions that are well spread over the Pareto front in all the runs of all test problems. In addition, the results of the algorithm are compared with some of the constrained multiobjective algorithms suggested so far. The comparison results indicate that the proposed algorithm performs better than the other algorithms in that it is able to provide a well distributed and consistent Pareto front that has optimal individuals. For future work, the authors recommend applying the proposed constraint handling technique using modified objective function formulation to other multiobjective evolutionary approaches, such as SPEA2 [41]. Since the proposed constraint handling technique is very generic, the modification of fitness function through distance and adaptive penalty measure can be easily realized for most MOEAs and MOPSOs [42]. In addition, the proposed algorithm can be directly applied to solve real-world problems, such as [43] and [44].

REFERENCES

[1] B. Tessema and G. G. Yen, "A self-adaptive constrained evolutionary algorithm," in *Proc. IEEE Cong. Evol. Comput.*, Vancouver, Canada, 2006, pp. 246–253.

[2] H. Lu and G. G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 325–343, 2003.

[3] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, 2003.

[4] C. A. Coello Coello, "Evolutionary multiobjective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, 2006.

[5] C. A. Coello Coello, EMOO Repository Webpage. [Online]. Available: http://delta.cs.cinvestav.mx/~ccoello/EMOO/

[6] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization: Formulation, discussion, and generalization," in *Proc. Int. Conf. Genetic Algorithms*, Urbana-Champaign, IL, 1993, pp. 416–423.

[7] N. Srinivas and K. Deb, "Multi-objective function optimization using non-dominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, 2002.

[9] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[10] E. Zitzler and L. Thiele, "An evolutionary algorithm for multi-objective optimization: The strength Pareto approach," Swiss Federal Inst. Technol. (ETH), Switzerland Comput. Eng. and Networks Lab. (TIK), Tech. Rep..

[11] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 12, pp. 80–92, 2008.

[12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, pp. 770–784, 2007.

[13] T. Bäck, F. Hoffmeister, and H. Schwefel, "A survey of evolution strategies," in *Proc. Int. Conf. Genetic Algorithms*, San Diego, CA, 1991, pp. 2–9.

[14] A. Homaifar, S. H. Y. Lai, and X. Qi, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–254, 1994.

[15] J. Joines and C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs," in *Proc. Congr. Evol. Comput.*, Orlando, FL, 1994, pp. 579–584.

[16] J. C. Bean and A. B. Alouane, A dual genetic algorithm for bounded integer programs Univ. Michigan, Dept. Industrial and Operations Engineering, Ann Arbor, MI, Tech. Rep. TR 92-5, 1992, .

[17] R. Farmani and J. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 445–455, 2003.

[18] K. Deb, "An efficient constraint handling methods for genetic algorithms," *Comput. Methods in Appl. Mech. Eng.*, vol. 186, pp. 311–338, 2000.

[19] T. P. Runarsson and X. Yao, "Stochastic ranking for constraint evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 284–294, 2000.

[20] T. P. Runarsson and X. Yao, "Search bias in constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern.*, vol. 35, pt. C, pp. 233–243, 2005.

[21] T. Takahama and S. Sakai, "Constrained optimization by applying the α-constrained method to the nonlinear simplex method with mutations," *IEEE Trans. Evol. Comput.*, vol. 9, pp. 437–451, 2005.

[22] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elite," in *Proc. IEEE Cong. Evol. Comput.*, Vancouver, Canada, 2006, pp. 308–315.

[23] Y. Wang and Z. Cai, "A multi-objective optimization based evolutionary algorithm for constrained optimization," in *Proc. IEEE Cong. Evol. Comput.*, Edinburgh, U.K., 2005, pp. 1081–1087.

[24] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, pp. 424–435, 2005.

[25] J. Aidanpaa, J. Anderson, and A. Angantyr, "Constrained optimization based on a multi-objective evolutionary algorithm," in *Proc. Cong. Evol. Comput.*, Canberra, Australia, 2003, pp. 1560–1567.

[26] C. A. Coello Coello and A. D. Christiansen, "MOSES: A multi-objective optimization tool for engineering design," *Eng. Opt.*, vol. 31, no. 3, pp. 337–368, 1999.

[27] T. T. Binh and U. Korn, "MOBES: A multi-objective evolution strategy for constrained optimization problems," in *Proc. Int. Conf. Genetic Algorithms*, East Lansing, MI, 1997, pp. 176–182.

[28] F. Jimenez, A. F. Gomez-Skarmeta, G. Sanchez, and K. Deb, "An evolutionary algorithm for constrained multi-objective optimization," in *Proc. IEEE Cong. Evol. Comput.*, Honolulu, HI, 2002, pp. 1133–1138.

[29] T. Ray, K. Tai, and K. C. Seow, "An evolutionary algorithm for multi-objective optimization," *Eng. Opt.*, vol. 33, no. 3, pp. 399–424, 2001.

[30] D. Chafekar, J. Xuan, and K. Rasheed, "Constrained multi-objective optimization using steady state genetic algorithms," in *Proc. Genetic and Evol. Comput. Conf.*, Chicago, IL, 2003, pp. 813–824.

[31] N. Young, "Blended ranking to cross infeasible regions in constrained multi-objective problems," in *Proc. Int. Conf. Comput. Intell. Modeling, Control and Automation, and Int. Conf. Intell.Agents, Web Technologies and Internet Commerce*, Sydney, Australia, 2005, pp. 191–196.

[32] A. Angantyr, J. Andersson, and J. Aidanpaa, "Constrained optimization based on a multi-objective evolutionary algorithm," in *Proc. Cong. Evol. Comput.*, Canberra, Australia, 2003, pp. 1560–1567.

[33] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms- Part I: A unified formulation," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 26–37, 1998.

[34] K. Harada, J. Sakuma, I. Ono, and S. Kobayashi, "Constraint-handling method for multi-objective function optimization: Pareto descent repair operator," in *Proc. Int. Conf. Evol. Multi-Criterion Opt.*, Matshushima, Japan, 2007, pp. 156–170.

[35] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[36] V. Chankong and Y. Y. Haimes, *Multi-Objective Decision Making Theory and Methodology*. New York: North-Holland, 1983.

[37] A. Osyczka and S. Kundu, "A new method to solve generalized multi-criteria optimization problems using the simple genetic algorithm," *Structural Opt.*, vol. 10, no. 2, pp. 94–99, 1995.

[38] M. Tanaka, "GA-based decision support system for multi-criteria optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Opt.*, Guanajuato, Mexico, 1995, pp. 1556–1561.

[39] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 117–132, 2003.

[40] J. D. Knowles, L. Thiele, and E. Zitzler, A tutorial on performance assessment of stochastic multiobjective optimizers Swiss Federal Inst. Technol. (ETH), Switzerland Comput. Eng. Networks Lab. (TIK), Munich, Switzerland, TIK-Rep. 214, revised, 2006, .

[41] E. Zitzler, M. Laumanns, and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm Swiss Federal Inst. Technol. (ETH), Switzerland Comput. Eng. Networks Lab. (TIK), Munich, Switzerland, Tech. Rep. 103, 2001.

[42] W. F. Leong and G. G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1270–1293, Oct. 2008.

[43] P. G. de Lima and G. G. Yen, "Multiple objective evolutionary algorithm for temporal linguistic rule extraction," *ISA Trans.*, vol. 44, no. 2, pp. 315–327, Apr. 2005.

[44] M. L. Goldstein and G. G. Yen, "Using evolutionary algorithms for defining the sampling policy of complex n-partite networks," *Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 762–773, Jun. 2005.

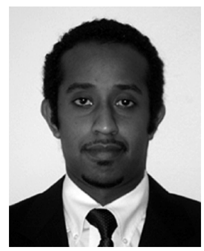
Yonas Gebre Woldesenbet received the B.S. degree in electrical engineering from Bahir Dar University, Bahir Dar, Ethiopia, in 2004 and the M.S. degree in electrical engineering from Oklahoma State University, Stillwater, in 2007, and received the Gold-Medal Award for the most outstanding academic achievement at Bahir Dar University.

He is currently a Software Engineer. His research interest comprises of a wide variety of subject matters including dynamic evolutionary optimization, constraint handling in multiobjective optimization, intelligent systems, VLSI, and mathematical modeling and analysis.

Gary G. Yen (S'87–M'88–SM'97–F'09) received the Ph.D. degree in electrical and computer engineering from the University of Notre Dame, Notre Dame, IN, in 1992.

He is currently a Professor at the School of Electrical and Computer Engineering, Oklahoma State University (OSU), Stillwater. Before joining OSU in 1997, he was with the Structure Control Division, U.S. Air Force Research Laboratory, Albuquerque, NM. His research is supported by the DoD, DoE, EPA, NASA, NSF, and Process Industry. His research interest includes intelligent control, computational intelligence, conditional health monitoring, signal processing and their industrial/defense applications.

Dr. Yen was an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, the *IEEE Control Systems Magazine*, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, *Automatica*, and IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, PART A and PART B. He is currently serving as an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION AND MECHATRONICS. He served as the General Chair for the 2003 IEEE International Symposium on Intelligent Control held in Houston, TX, and 2006 IEEE World Congress on Computational Intelligence held in Vancouver, Canada. In addition, he served as Vice President for the Technical Activities of the IEEE Computational intelligence Society in 2005 and 2006, and is the founding Editor-in-Chief of the *IEEE Computational Intelligence Magazine* since 2006.



Biruk G. Tessema graduated with a degree in electrical engineering from Bahir Dar University, Bahir Dar, Ethiopia, in 2004. He received the M. S. degree in electrical engineering from Oklahoma State University, Stillwater, in 2006.

His research interest includes computational intelligence, constraint handling, multiobjective optimization, and optimization and economic theory applied to electric power system operations.