An Adaptive Tradeoff Model for Constrained Evolutionary Optimization

Yong Wang, Zixing Cai, *Senior Member, IEEE*, Yuren Zhou, and Wei Zeng

***Abstract*—In this paper, an adaptive tradeoff model (ATM) is proposed for constrained evolutionary optimization. In this model, three main issues are considered: 1) the evaluation of infeasible solutions when the population contains only infeasible individuals; 2) balancing feasible and infeasible solutions when the population consists of a combination of feasible and infeasible individuals; and 3) the selection of feasible solutions when the population is composed of feasible individuals only. These issues are addressed in this paper by designing different tradeoff schemes during different stages of a search process to obtain an appropriate tradeoff between objective function and constraint violations. In addition, a simple evolutionary strategy (ES) is used as the search engine. By integrating ATM with ES, a generic constrained optimization evolutionary algorithm (ATMES) is derived. The new method is tested on 13 well-known benchmark test functions, and the empirical results suggest that it outperforms or performs similarly to other state-of-the-art techniques referred to in this paper in terms of the quality of the resulting solutions.**

***Index Terms*—Constrained optimization, evolutionary strategy (ES), multiobjective optimization, tradeoff model.**

I. Introduction

EVOLUTIONARY ALGORITHMs (EAs) have been broadly applied to tackle global optimization problems. Over the past decade, solving constrained optimization problems (COPs) via EAs has attracted much attention. By integrating various constraint-handling techniques with EAs, researchers have proposed a large number of constrained optimization evolutionary algorithms (COEAs) ([1]–[3]).

Without loss of generality, the nonlinear programming (NLP) problem of interest can be formulated as follows (in minimization sense):

$$\text{Find } \vec{x}(\vec{x} = (x_1, x_2, \ldots, x_n) \in \Re^n) \quad \text{such that } f(\vec{x}) \text{ is optimized}$$

where $\vec{x} \in \Omega \subseteq S$, and S is an n-dimensional rectangle space in $\Re^n$ defined by the parametric constraints

$$l(i) \le x_i \le u(i), \quad 1 \le i \le n.$$

The feasible region $\Omega \subseteq S$ is defined by a set of m additional linear or nonlinear constraints $(m \ge 0)$

$$g_j(\vec{x}) \le 0, \ j = 1, \ldots, q \ \text{ and } \ h_j(\vec{x}) = 0, \ j = q+1, \ldots, m$$

where q is the number of inequality constraints and $m - q$ is the number of equality constraints.

If an inequality constraint that satisfies $g_j(\vec{x}) = 0 (j \in \{1, \ldots, q\})$ at any point $\vec{x} \in \Omega$, we say it is active at $\vec{x}$. All equality constraints $h_j(\vec{x}) (j = q+1, \ldots, m)$ are considered *active* at all points of Ω.

As we know, the ultimate goal of COEAs is to find the feasible optimal solution. To achieve this goal, more feasible individuals are required to be involved in reproduction during the evolutionary process. However, on the other hand, some infeasible individuals may carry more important information for the final solution than their feasible counterparts in some generations. Hence, these two aspects lead to a contradiction in constrained evolutionary optimization. To address this contradiction, the main challenge is to handle the constraints and to optimize the objective function simultaneously. One possible way is to determinate the tradeoff between the constraint violations and the objective function. Although various tradeoff mechanisms have been proposed, adaptive tradeoff has seldom been investigated.

In this paper, we propose a novel *adaptive tradeoff model* (ATM) for constrained evolutionary optimization. This model takes advantage of the valuable information derived from the last evolution to direct the next evolution. In general, a constraint-handling technique will inevitably encounter the following three stages in the process of search: 1) the population contains infeasible individuals only; 2) the population consists of both feasible and infeasible individuals; and 3) the population is entirely composed of feasible individuals. In this paper, the approach is to design a tradeoff scheme in each stage according to the characteristic of each stage.

In the first stage, we design a hierarchical nondominated individual selection scheme, which tends to guide the population toward feasibility from various directions. At the second stage, we devise a converted fitness function to adaptively balance feasible and infeasible solutions according to the information provided by the last population. In this way, a diverse and robust search is guaranteed. Finally, during the third stage, the selection of individuals is based solely on their objective function values,

Manuscript received March 2, 2006; revised July 23, 2006, October 15, 2006, and January 8, 2007. This work was supported in part by the National Natural Science Foundation of China under Grant 60234030, Grant 60673062, and Grant 60404021, in part by the National Basic Scientific Research Founds under Grant A1420060159, in part by Hunan S&T Founds under Grant 05IJY3035, and in part by the Natural Science Foundation of Guangdong Province of China under Grant 06025686.

Y. Wang, Z. Cai, and W. Zeng are with the College of Information Science and Engineering, Central South University, Changsha, Hunan, China (e-mail: ywang@csu.edu.cn; zxcai@csu.edu.cn).

Y. Zhou is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong, China (e-mail: yrzhou@scut.edu.cn).

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.2007.902851

since all of them are feasible in this case. These three schemes in different stages constitute the ATM in this paper. Moreover, a generic COEA (ATMES) is obtained by integrating ATM with evolutionary strategy (ES) which is an important part of EAs.

We evaluate the efficiency and effectiveness of the proposed method on 13 benchmark test functions. The experimental results indicate that ATMES is competitive with, or superior to, some state-of-the-art COEAs, such as stochastic ranking (SR) [4] and simple multimembered evolutionary strategy (SMES) [5], when measured by several performance metrics, e.g., the best, median, mean, and worst objective function values, and the standard deviations.

The remainder of this paper is organized as follows. Section II presents a detailed review of tradeoff methods in constrained evolutionary optimization. In Section III, the proposed ATM, which involves three independent phases is presented. The experimental results are reported in Section IV. In Section V, the effects of algorithm parameters on the performance are demonstrated by various experiments. Finally, Section VI concludes this paper.

II. Tradeoff in Constrained Evolutionary Optimization

A great deal of work has already been undertaken on constraint-handling techniques, which are the main focus of this paper. Michalewicz and Schoenauer [1] and Coello [2] provided a comprehensive survey of the most popular constraint-handling techniques currently used with EAs, grouping them into four and five categories, respectively. As stated in [2], the constraint-handling techniques can be divided into five categories: 1) penalty functions; 2) special representations and operators; 3) repair algorithms; 4) separate objective and constraints; and 5) hybrid methods. Next, we will review these techniques from the tradeoff point-of-view.

Penalty function methods are among the most common methods to solve COPs. The principal idea of these methods is to transform a COP into an unconstrained one by introducing a penalty term into the original objective function in order to penalize constraint violations. In general, a penalty term is based on the degree of constraint violation of an individual. Let

$$G_j(\vec{x}) = \begin{cases} \max\{0, g_j(\vec{x})\}, & 1 \le j \le q \\ \max\{0, |h_j(\vec{x})| - \varepsilon\}, & q+1 \le j \le m \end{cases} \quad (1)$$

where ε is a positive tolerance value for inequality constraints. Then, $G(\vec{x}) = \sum_{j=1}^{m} G_j(\vec{x})$ reflects the degree of constraint violation of the individual $\vec{x}$.

As analyzed in [4], all that a penalty method tries to do is to obtain the right tradeoff between the objective function and the penalty function so that the search moves toward the optimum in the feasible space. In [4], the authors characterized the problem of choosing the appropriate coefficient r_g for the penalty function, describing how it affected the domination between the constraint violations and the objective function when deciding the rank of each individual. Indeed, for any population, there exists a certain range $[r_1, r_2]$, such that: 1) if $r_g < r_1$, then the comparisons of individuals are based solely on their objective function; 2) if $r_g > r_2$, then the comparisons of individuals are based solely on their penalty function; and 3) if $r_g \in [r_1, r_2]$, then the comparisons of individuals are based on a combination of their objective and penalty functions. Notice that the values of the parameters r_1 and r_2 are related to the given population and, consequently, they are problem dependent.

Adaptive penalty methods are very promising for constrained optimization, since they can make use of information obtained during the search to adjust their own parameters. Rasheed [6] proposed an adaptive penalty approach for constrained genetic-algorithm optimization. The idea is to start with a relatively small penalty coefficient and then increase it or decrease it on demand as the optimization progresses. Farmani and Wright [7] presented an adaptive fitness formulation method, an enhanced version of the algorithm in [8], where the penalty is divided into two stages. The improved approach eliminates the fixed weight for the second penalty stage proposed in [8], by assigning the penalized objective function value of the worst infeasible individual to be equal to that of the individual with maximum objective function value in the current population. This makes the method adaptive and more dynamic due to the fact that the individual with maximum objective function value may vary from one generation to another.

In [4], Runarsson and Yao also introduced a SR-based method to balance the objective and penalty functions. A probability parameter p_f is involved to compare individuals as follows: given pairwise adjacent individuals: 1) if both individuals are feasible, the one with better objective function value wins, else 2) if a uniformly generated random number u between 0 and 1 is less than p_f, the one with better objective function value wins, otherwise, the one with a smaller degree of constraint violation wins. This approach significantly improves the search performance without any special constrained operator.

Definition 1 (Pareto Dominance): A vector $\vec{u} = (u_1, \ldots, u_k)$ is said to *Pareto dominate* another vector $\vec{v} = (v_1, \ldots, v_k)$, denoted as $\vec{u} \prec \vec{v}$, if

$$\forall i \in \{1, \ldots, k\}, u_i \le v_i \quad \text{and} \quad \exists j \in \{1, \ldots, k\}, u_j < v_j.$$

In terms of the SR, alternative selection criteria can be employed to analyze it: given pairwise adjacent individuals: 1) if one individual Pareto dominates the other, the superior one is preferred; else, namely, the two individuals are nondominated with respect to each other and 2) if $u < p_f$, the one with better objective function value is preferred, else the one with smaller constraint violation is preferred. It is provable that the above criteria are equivalent to the SR. However, from these criteria, we can make the SR easier to understand. It is obvious that the probability p_f plays an important role only when the two individuals in comparison are nondominated with each other.

Inspired by Powell and Skolnick [9], Deb [10] devised a fitness formulation as follows:

$$F(\vec{x}) = \begin{cases} f(\vec{x}), & \text{if } g_j(\vec{x}) \le 0, \forall j = 1, 2, \ldots, m \\ f_{\max} + \sum_{j=1}^{m} G_j(\vec{x}), & \text{otherwise} \end{cases} \quad (2)$$

where $f_{\max}$ is the objective function value of the worst feasible solution in the population. This method uses a tournament selection operator, i.e., two solutions are compared at a time. When comparing pairwise solutions, the fitness formulation in (2) reflects the following comparison criteria: 1) any feasible solution

is preferred to any infeasible solution; 2) between two feasible solutions, the one with better objective function value is preferred; and 3) between two infeasible solutions, the one with smaller degree of constraint violation is preferred. Such tournament selection can be regarded as a special case of SR when p_f is equal to 0. The disadvantage of this method is that enough emphasis has not been placed on infeasible individuals. If a large number of individuals in the population are feasible, then infeasible individuals in the current generation have little chance to survive and, therefore, this method has a strong tendency to get stuck in a local optimum. Based on this method, Mezura and Coello [5] introduced a simple diversity mechanism in order to introduce some infeasible solutions into the next population.

Takahama and Sakai [11] proposed the α constrained method that converts an algorithm for unconstrained optimization problems into an algorithm for COPs by replacing ordinary comparisons with the α level comparison. In this approach, the authors apply the following tradeoff to the objective function and the constraint violations: given pairwise individuals x_1 and x_2, let u_1 and u_2 denote their level of satisfaction of the constraints, respectively: 1) if $u_1, u_2 \geq \alpha$, then the one with better objective function value wins; 2) if $u_1 = u_2$, then the one with better objective function value wins; otherwise; 3) the one with higher satisfaction level wins, where the α level is controlled according to an exponential function and $0 \leq \alpha \leq 1$.

Using multiobjective optimization to tackle COPs can be considered as another kind of tradeoff in constrained evolutionary optimization, since it is essentially desired to also balance the objective function and the constraint violations. In this case, constraints can be regarded as one or more objectives. If constraints are treated as one objective, then the original problem will be transformed into a multiobjective optimization problem (MOP) that has two objectives. In general, one objective is the original objective function $f(\vec{x})$, and the other one is the degree of constraint violation of an individual $\vec{x}$, i.e., $G(\vec{x})$. Alternatively, each constraint can be treated as an objective and the original problem is transformed into a MOP which has $m + 1$ objectives. Therefore, we get a new vector $\vec{v} = (f(\vec{x}), f_1(\vec{x}), \ldots, f_m(\vec{x}))$ to be optimized, where $f_1(\vec{x}), \ldots, f_m(\vec{x})$ are the constraints of the given problem. Several typical paradigms are reviewed next.

By combining the vector evaluated genetic algorithm (VEGA) [12] with Pareto ranking, Surry and Radcliffe proposed a method called COMOGA [13]. COMOGA uses a single population but randomly decides whether to consider the original problem as a constraint satisfaction problem or as an unconstrained optimization problem. The relative likelihood of adopting each view is adjusted using a simple adaptive mechanism that tries to set a target proportion (e.g., 0.1) of feasible solutions in the population. Its main advantage is that it does not require a tuning of penalty factor.

Venkatraman and Yen [14] presented a two-phase framework to solve COPs. In the first phase, COP is treated as a constrained satisfaction problem, as in [13], and the genetic search is directed toward minimizing the constraint violations of the solutions. In the second phase, COP is treated as a biobjective optimization problem, and a nondominated sorting algorithm, as described in [15], is adopted to rank the individuals. This algorithm has the advantage of being problem independent and has the added benefit of not having to rely on any parameter tuning.

Zhou *et al.* [16] proposed a novel method which uses Pareto dominance to assign an individual's Pareto strength.

Definition 2 (Pareto Strength): Each individual $\vec{x}_i$ in a population P_t is assigned an integer $s(\vec{x}_i)$, called Pareto strength of $\vec{x}_i$. $s(\vec{x}_i)$ is the number of individuals in the population P_t Pareto dominated by $\vec{x}_i$, that is

$$s(\vec{x}_i) = \#\{\vec{x}_j | \vec{x}_j \in P_t \wedge \vec{x}_i \prec \vec{x}_j\}$$

where $\#$ is the cardinality of the set.

Based on Pareto strength, the ranking of individuals in the population is conducted in such a manner that, upon comparing each individual's strength: 1) the one with higher strength wins and 2) if the strength is equal, the one with lower degree of constraint violation is better.

Besides, other methods are also developed to handle COPs, for example, the method [17] based on population-based multiobjective technique such as VEGA, the method [18] based on Fonseca and Fleming's Pareto ranking process [19], the method [20] based on the niched-Pareto genetic algorithm (NPGA) [21], and the method [22] based on the Pareto archived evolutionary strategy (PAES) [23].

From our analyses of the algorithms previously proposed to solve COPs, we can conclude that their fundamental principle is to design an appropriate tradeoff between the objective function and the constraint violations. However, the method describing how to make the tradeoff adaptive is usually neglected by most of them, namely, most of them are not capable of exploiting the helpful information acquired during the evolution to guide the population for further search. Motivated by this consideration, this paper proposes an ATM that consists of three main phases. Furthermore, a generic COEA (i.e., ATMES) is obtained by integrating a simple ES with the ATM.

III. An Adaptive Tradeoff Model (ATM)

As previously discussed, in general, a constraint-handling technique will inevitably experience three phases. Therefore, we can develop alternate tradeoff schemes for different phases to facilitate a more explicit adaptation. Next, we will discuss how to design such tradeoff schemes step by step.

A. Phase One

Phase one refers to the case where the current population contains no feasible solutions, namely, $fp = 0$, where fp represents the feasibility proportion of the current population.

In this phase, while the feasibility of an individual $\vec{x}$ is more important than the minimization of its objective function $f(\vec{x})$, the diversity in the population should also be considered carefully. A desirable search mechanism should guide the population toward feasibility from various directions (see Fig. 1), since we have no *a prior* knowledge about the location of the global minimum.

After observing the fact that some methods (such as the SR) simultaneously handle both objective functions and constraints, but fail to produce feasible solutions for every run, Venkatraman and Yen [14] argued that the objective function should be com-

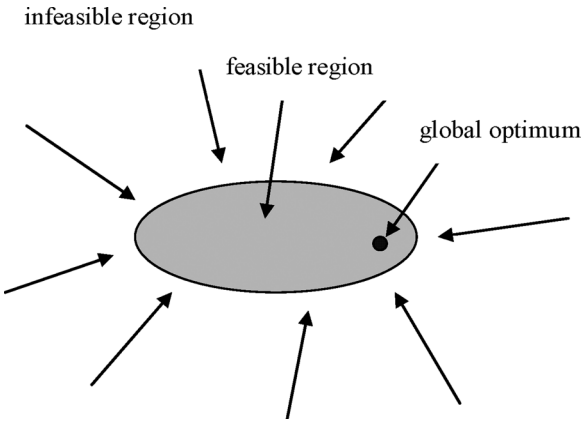


Fig. 1. A search schematic diagram.

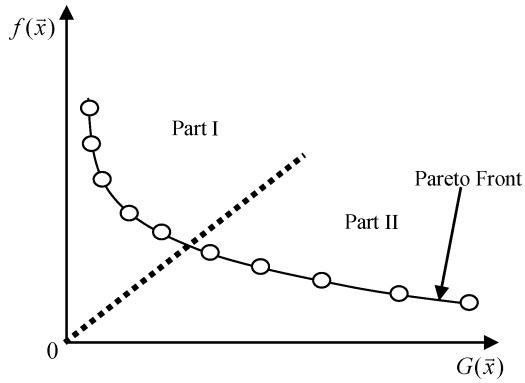


Fig. 2. Schematic diagram to illustrate the hierarchical nondominated individual selection scheme. The individuals in Part I will be selected and deleted from the population subsequently, since the constraint violations of them are less than those of the individuals in Part II. In addition, the individuals in Part II will remain in the remaining population for the next competition.

pletely disregarded in the first phase so as to provide a greater assurance of producing feasible solutions. Nevertheless, such a mechanism not only lacks the desirable property of being able to attain low-fitness, nearly feasible solutions, but may also be a poor method of searching the feasible region if the feasible region is a highly sparse and disconnected subset of the search space. Therefore, constraint satisfaction certainly needs to be reconciled with the optimization of the objective function in the first stage; but some special operators to bias the search are completely necessary.

In our tradeoff scheme for this phase, a hierarchical nondominated individual selection scheme is proposed. Since this scheme is based on Pareto dominance, three related definitions in multiobjective optimization are given as follows. For the sake of clarity, let $\mathbf{f}(\vec{x}) = (f(\vec{x}), G(\vec{x}))$.

Definition 3 (Pareto Optimality): $\vec{x}_u \in S$ is said to be *Pareto optimal* (in S) if and only if $\neg\exists\vec{x}_v \in S$, $\vec{v} \prec \vec{u}$, where $\vec{v} = \mathbf{f}(\vec{x}_v) = (v_1, v_2)$, $\vec{u} = \mathbf{f}(\vec{x}_u) = (u_1, u_2)$.

Definition 4 (Pareto Optimal Set): The *Pareto optimal set*, denoted as ρ^* is defined as

$$\rho^* = \{\vec{x}_u \in S | \neg\exists\vec{x}_v \in S, \vec{v} \prec \vec{u}\}.$$

The vectors included in the Pareto optimal set are called *nondominated individuals*.

Definition 5 (Pareto Front): According to the Pareto optimal set, the *Pareto front*, denoted as ρf^* is defined as

$$\rho f^* = \{\vec{u} = \mathbf{f}(\vec{x}_u) | \vec{x}_u \in \rho^*\}.$$

Clearly, the Pareto front is the image of the Pareto optimal set in the objective space.

In the hierarchical scheme proposed, since nondominated individuals represent the Pareto optimal set of the population, they will be identified in the population as the prior candidates.

Intuitively, with respect to constrained optimization, it does not make sense if an individual is far away from the boundaries of the feasible region. Thus, we are only concerned with those nondominated individuals with less constraint violations in the population. Note that this can act as a search bias. A simple way to achieve this is to select only the first half of nondominated individuals and to store them into the offspring population, after ranking nondominated individuals based on their constraint violations in ascending order. An illustration is shown in Fig. 2. Subsequently, the selected individuals are deleted from the parent population. Next, half of nondominated individuals with less constraint violations in the remaining population are also stored into the offspring population, and then eliminated from the parent population. This process continues until the number of individuals archived reaches the size of the offspring population.

A property of the process above is given as follows.

Property 1: The individual with the minimum constraint violation in the parent population is the first individual in the offspring population.

This scheme tends to guide the population toward feasibility from various directions by constantly partitioning the hierarchy of individuals in the population.

Remark 1: The second half of nondominated individuals in a nondominated level is given an opportunity to remain in the rest of the population for the next competition. The reason is that these individuals may also contain some important information and have minor constraint violations, even if their constraint violations are greater than those of the first half of nondominated individuals at the same nondominated level. Indeed, if these individuals are certain to have a relatively higher degree of constraint violations, the possibility that they are selected and archived into the offspring population is low, even though they are kept in the remaining population.

Remark 2: Although in [24] nondominated individuals are also concerned when the population is infeasible; the identification of nondominated individuals is only based on the constraint violations.

B. Phase Two

Phase two refers to the case where the current population consists of a combination of feasible and infeasible individuals, namely, $0 < fp < 1$.

In this phase, it is commonly accepted that the infeasible individuals with better objective function values and lower infeasibility should survive into the next population. The reason is that such individuals may be more important than their feasible counterparts in some generations, especially when the proportion of the feasible region is very small compared to the entire

search space or the optimum is located exactly on the boundaries of the feasible region. However, on the other hand, since the ultimate goal of COEAs is to find the feasible optimal solution, we cannot forsake feasible solutions absolutely. In general, we tend to retain some important feasible and infeasible solutions in the population.

The second tradeoff scheme is proposed to accomplish the above goal. It has the capability to adaptively convert the fitness of individuals based on the feasibility proportion of the last population, and works as follows. Suppose that a population has λ individuals, whose subscripts are recorded into a set Z, i.e., $\mathrm{Z} = \{1, 2, \ldots, \lambda\}$. Then, the population is divided into the feasible group Z_1 and the infeasible group Z_2, according to the feasibility of each individual

$$\mathrm{Z}_1 = \arg\{\bar{x}_i | G(\bar{x}_i) = 0, i \in \mathrm{Z}\} \tag{3}$$

$$\mathrm{Z}_2 = \arg\{\bar{x}_i | G(\bar{x}_i) > 0, i \in \mathrm{Z}\}. \tag{4}$$

Also, the best and worst feasible solutions are found by (5) and (6)

$$f_{\min} = \min_{i \in \mathrm{Z}_1} f(\vec{x}_i) \tag{5}$$

$$f_{\max} = \max_{i \in \mathrm{Z}_1} f(\vec{x}_i). \tag{6}$$

The converted objective function $f'(\vec{x}_i)(i \in \mathrm{Z})$ has the following form:

$$f'(\vec{x}_i) = \begin{cases} f(\vec{x}_i), & i \in \mathrm{Z}_1 \\ \max\{\varphi * f_{\min} + (1-\varphi) * f_{\max}, f(\vec{x}_i)\}, & i \in \mathrm{Z}_2 \end{cases} \tag{7}$$

where φ denotes the feasibility proportion of the last population. Note that only the objective function of infeasible individuals is altered after conversion. Each objective function value is then normalized

$$f_{\mathrm{nor}}(\vec{x}_i) = \frac{f'(\vec{x}_i) - \min_{j \in \mathrm{Z}} f'(\vec{x}_j)}{\max_{j \in \mathrm{Z}} f'(\vec{x}_j) - \min_{j \in \mathrm{Z}} f'(\vec{x}_j)}, \quad i \in \mathrm{Z}. \tag{8}$$

Similarly, the constraint violations can be normalized according to

$$G_{\mathrm{nor}}(\vec{x}_i) = \begin{cases} 0, & i \in \mathrm{Z}_1 \\ \dfrac{G(\vec{x}_i) - \min\limits_{j \in \mathrm{Z}_2} G(\vec{x}_j)}{\max\limits_{j \in \mathrm{Z}_2} G(\vec{x}_j) - \min\limits_{j \in \mathrm{Z}_2} G(\vec{x}_j)}, & i \in \mathrm{Z}_2 \end{cases}. \tag{9}$$

Notice that the normalized constraint violations of infeasible individuals are based only on the maximum and minimum constraint violations in the infeasible group. In contrast, the normalization of objective function value is based on both the feasible and infeasible groups.

A final fitness function is obtained by adding the normalized objective function and constraint violations together

$$f_{\mathrm{final}}(\vec{x}_i) = f_{\mathrm{nor}}(\vec{x}_i) + G_{\mathrm{nor}}(\vec{x}_i), \quad i \in \mathrm{Z}. \tag{10}$$

As for the above transformation, it has several important properties summarized as follows.

Property 2: The comparisons among feasible solutions are based only on their objective function values. However, both the objective function values and the degree of constraint violations should be taken into account when comparing feasible solutions with infeasible solutions or comparing among infeasible solutions.

Property 3: If the parameter φ has a larger value, the objective function values of infeasible individuals defined by (7) are smaller, thereby the probability increases for infeasible individuals to survive into the next population. On the contrary, if the parameter φ has a lower value, the objective function values of infeasible individuals defined by (7) are greater, which induces feasible solutions to be selected with a higher probability. Obviously, these behaviors reflect the adaptive feature of our approach.

Property 4: Some infeasible individuals with lower objective function values and lower infeasibility are considered better than some feasible ones.

Property 5: The best feasible individual in the current population is also the individual with the minimum fitness[1]; this ensures that the best individual in the current population is always feasible.

Due to these good properties, this tradeoff scheme is an effective approach to achieve the goal in the second phase.

Now, we employ an example to illustrate the above process. Suppose that a population contains ten individuals with the following vector values of $\mathbf{f}(\vec{x})$:

$$\mathbf{f}(\vec{x}_1) = (1, 0); \mathbf{f}(\vec{x}_2) = (1.5, 0); \mathbf{f}(\vec{x}_3) = (2, 0);$$
$$\mathbf{f}(\vec{x}_4) = (2.5, 0); \mathbf{f}(\vec{x}_5) = (3, 0); \mathbf{f}(\vec{x}_6) = (0.5, 0.01);$$
$$\mathbf{f}(\vec{x}_7) = (1.2, 0.03); \mathbf{f}(\vec{x}_8) = (1.7, 0.005);$$
$$\mathbf{f}(\vec{x}_9) = (2.2, 0.001); \mathbf{f}(\vec{x}_{10}) = (3.2, 0.02).$$

The first five individuals are feasible and the remaining individuals are infeasible. With respect to the different values of φ, we sort the ten participant individuals in ascending order of their fitness.

If $\varphi = 0.0$, the corresponding ordered sequence is

$$1,\ 2,\ 3,\ 4,\ 5,\ 9,\ 8,\ 6,\ 10,\ 7.$$

Apparently, a preference has been given to feasible solutions in this case so that the next population may involve more feasible solutions.

If $\varphi = 0.5$, the corresponding ordered sequence becomes

$$1,\ 2,\ 3,\ 9,\ 8,\ 4,\ 6,\ 5,\ 7,\ 10.$$

If $\varphi = 1.0$, the corresponding ordered sequence becomes

$$1,\ 2,\ 6,\ 3,\ 8,\ 9,\ 4,\ 5,\ 7,\ 10.$$

Regarding these two cases, some infeasible solutions can outperform some feasible solutions, so that a proper balance between feasible and infeasible solutions might always be maintained. Moreover, the larger the value of the parameter φ, the higher the probability that infeasible solutions can survive into the next population.

From the above results, one can conclude that the tradeoff scheme in phase two is capable of adapting the rank of feasible

[1]Note that the algorithm is formulated for solving minimization problems.

and infeasible solutions by making use of the information obtained from the last population.

Remark 3: Farmani and Wright [7] used the information of the best individual and the worst infeasible individual to calculate the fitness of a solution. However, the main idea of this method is adaptive fitness formulation and it does not consider the feasibility proportion in the population. Deb [10] also adopted a converted fitness to select individuals, but in his approach, infeasible solutions are always ranked below feasible solutions for selection in the reproduction process. In addition, although Surry and Radcliffe [13] used an adaptively probability $p_{\cos t}$ to determine the likelihood of selection based on objective function value (in this case the remaining individuals are selected based on Pareto ranking with respect to constraint violations), the adjusting of the parameter $p_{\cos t}$ is done by setting a fixed target proportion of feasible solutions in the population (e.g., 0.1).

C. Phase Three

Phase three refers to the case where the current population is entirely composed of feasible individuals, namely, $fp = 1$.

It is obvious that the evolution of this phase is totally equivalent to that of unconstrained optimization. Thus, the comparisons of individuals are based only on their objective function values. This essentially is also a tradeoff, since the fitness function can be formulated in the following form:

$$f_{\text{final}}(\vec{x}_i) = f(\vec{x}_i) + G(\vec{x}_i), \quad i \in \mathrm{Z} \tag{11}$$

where the term $G(\vec{x})$ has no influence on the individual evaluation.

We incorporate the individual components described above and present ATM in Fig. 3.

D. Computational Time Complexity

Consider the complexity for one iteration of ATMES. Note that we use a simple (μ, λ)-ES as the search algorithm that does not guarantee the globally optimal solution and will be specified later. In the entire algorithm, the basic operations and their worst-case complexities are as follows:

1) in the first phase, the hierarchical nondominated individual selection scheme is $O(2\lambda^2)$;
2) in the second and third phases, the sorting based on (10) and (11) is $O(\lambda \log \lambda)$.

So, the overall complexity of the algorithm is $O(\lambda^2)$.

IV. Experimental Study

A. Test Functions and the Experimental Conditions

In this section, we apply the proposed ATM to 13 benchmark test functions from [4]. These test cases include various types (linear, nonlinear, and quadratic) of objective functions with different number of decision variables (n) and a range of types [linear inequalities (LI), nonlinear equalities (NE), and nonlinear inequalities (NI)], and number of constraints. The main characteristics of the test cases are reported in Table I, where a is the number of constraints active at the optimal solution. In addition, ρ is the approximated ratio between the size of the feasible search space and that of the entire search space, i.e.,

$$\rho = |\Omega|/|S| \tag{12}$$

where $|S|$ is the number of solutions randomly generated from S, $|\Omega|$ is the number of feasible solutions out of these $|S|$ solutions. In our experimental setup, $|S| = 1\ 000\ 000$.

In addition, we use a simple (μ, λ)-ES as the search engine, which is the identical version as in the SR [4] and only differs in the initial stepsize of the ES and the selection operator. There are two notable features of this kind of ES. First, it uses a global intermediate recombination applied only to the strategy parameters. Second, the variation of the objective parameters is retried if they fall outside of the parametric bounds. A mutation out of

function $ATM(A, A', fp, \varphi, \mu, \lambda)$

1. **Input:** $A = \{\vec{a}_1, \cdots, \vec{a}_\lambda\}$, $A' = \emptyset$, fp, φ, μ, λ
2. **if** $fp = 0$ /* phase one */
3. **while** $|A'| < \mu$ **do**
4. identify the nondominated individuals from the population A;
5. A'' ={the first half of nondominated individuals with less constraint violations};
6. $A = A \setminus A''$;
7. $A' = A' \cup A''$;
8. **end while**
9. A' ={the first μ individuals in A'};
10. **else if** $0 < fp < 1$ /* phase two */
11. A' ={the best μ individuals in the population A based on the equation (10)};
12. **else** /* phase three */
13. A' ={the best μ individuals in the population A based on the equation (11)};
14. **end if**
15. **Output:** A'

Fig. 3. Pseudocode of the proposed ATM.

TABLE I
SUMMARY OF 13 BENCHMARK FUNCTIONS

Fcn	n	Type of f	ρ	LI	NE	NI	a
g01	13	quadratic	0.0003%	9	0	0	6
g02	20	nonlinear	99.9965%	1	0	1	1
g03	10	nonlinear	0.0000%	0	1	0	1
g04	5	quadratic	26.9356%	0	0	6	2
g05	4	nonlinear	0.0000%	2	3	0	3
g06	2	nonlinear	0.0064%	0	0	2	2
g07	10	quadratic	0.0003%	3	0	5	6
g08	2	nonlinear	0.8640%	0	0	2	0
g09	7	nonlinear	0.5256%	0	0	4	2
g10	8	linear	0.0005%	3	0	3	3
g11	2	quadratic	0.0000%	0	1	0	1
g12	3	quadratic	0.0197%	0	0	9^3	0
g13	5	nonlinear	0.0000%	0	3	0	3

1. Initialize:
   $\vec{\sigma}_k^{'} = \breve{\sigma} * (\vec{u} - \vec{l}) / \sqrt{n}$, $k = 1, \cdots, \lambda$;
   Let, $\vec{x}_k^{'} = \vec{l} + (\vec{u} - \vec{l})\vec{U}(0,1)$, where, $\vec{u} = (u_1, \cdots, u_n)$ and $\vec{l} = (l_1, \cdots, l_n)$ are the upper and lower bounds of the decision variables respectively, n is the number of the decision variables;
   $\varphi = 0$; /* it means that feasible individuals always win the competition for reproduction with infeasible individuals in the first generation */
2. **while** termination criteria not satisfied **do**
3. evaluate : $f(\vec{x}_k^{'})$, $G(\vec{x}_k^{'})$, $k = 1, \cdots, \lambda$;
4. $fp = \dfrac{num\_1}{\lambda}$, where num_1 denotes the number of feasible solutions in the current population of size λ ;
5. select the best μ individuals from the λ points based on the ATM described in Fig. 3, i.e., $(\vec{x}_i, \vec{\sigma}_i) \leftarrow (\vec{x}_{i;\lambda}^{'}, \vec{\sigma}_{i;\lambda}^{'})$, $i = 1, \cdots, \mu$;
6. $\varphi = \dfrac{num\_2}{\mu}$, where num_2 denotes the number of feasible solutions in the current population of size μ ;
7. **for** $k = 1$ to λ **do** /* replication */
8. $i \leftarrow \mathrm{mod}(k - 1, \mu) + 1$; /* cycle through the best μ points */
9. $\sigma^{'}_{i,j} = (\sigma_{i,j} + \sigma_{k_j,j}) / 2$, $k_j \in \{1, \cdots, \mu\}$, $j = 1, \cdots, n$;
10. $\sigma^{'}_{k,j} \leftarrow \sigma^{'}_{i,j} \exp(\tau^{'} \mathrm{N}(0,1) + \tau \mathrm{N}_j(0,1))$, $j = 1, \cdots, n$;
11. $\vec{x}_k^{'} \leftarrow \vec{x}_i + \vec{\sigma}_k^{'} \vec{\mathrm{N}}(0,1)$; /* retry if out of bounds */
12. **end for**
13. **end while**

Fig. 4. Pseudocode of the proposed ATMES, where $\tau = (\sqrt{2\sqrt{n}})^{-1}$, $\tau' = (\sqrt{2n})^{-1}$, and k_j is a random number in $\{1, \ldots, \mu\}$.

bounds is retried only ten times after which it is set to its parent value. By combining ATM with this kind of ES, a generic COEA called ATMES is derived and depicted in Fig. 4.

For each test case, 30 independent runs are performed in MATLAB using the (50,300)-ES.[2] The number of generations is set to 800, thus the number of fitness function evaluations (FFEs) is equal to 240 000. Since the number of iterations used in this paper is nearly half of that used in SR, the initial stepsize of the ES in this paper is only 80% of that provided in SR (i.e., $\breve{\sigma} = 0.8$).[3]

In order to deal with equality constraints, each of them is converted into inequality constraints as $|h_j(\vec{x})| - \varepsilon \leq 0 (j = q + 1, \ldots, m)$, where ε is a small tolerance value. A dynamic setting of the parameter ε, which is originally proposed in ASCHEA [25] and used in [5] and [26] is adopted. The parameter ε decreases with respect to the current generation using the following expression:

$$\varepsilon(t+1) = \frac{\varepsilon(t)}{\breve{\varepsilon}}. \tag{13}$$

The initial ε_0 is set to 3,[4] and the value of $\breve{\varepsilon}$ is specified as $\breve{\varepsilon} = 1.0168$. Note that the use of the value 1.0168 causes the allowable tolerance for the equality constraints to go from 3 (initial value) to 5E-06 (final value), given the number of iterations adopted by our approach.

[2]The source code may be obtained from the authors upon request.

[3]The initial stepsize reduction is originally proposed in [5].

[4]Although the setting of the parameter ε_0 is based on experiments, we also refer to [5] when determining the value of it.

B. General Performance of the Proposed Algorithm

Table II summarizes the experimental results using the above parameters. This table shows the "known" optimal solutions for each test function and statistics for the 30 independent runs. These include the "best," "median," "mean," and "worst" objective function values, the standard deviations, and the average percentage of feasible solutions in the final population.

For each test function, the best solution is almost equivalent to the optimal solution. For test functions g01, g03, g04, g06, g08, and g11, the optimal solutions are consistently found in all 30 runs. For test functions g07, g09, g12, and g13, the near-optimal solutions are found in all 30 runs. For test function g02, the optimal solution is not consistently found, this benchmark function is known to have a very rugged fitness landscape and is, in general, the most difficult to solve. Another function whose optimum is not found consistently is g10; the main characteristic of this function is its relatively large search space. For test function g05, the near-optimal solution is found in 20 runs and the number of exceptions is only 10. Also, note that the standard deviations over 30 runs for all the test functions other than g10 are extremely small. This implies that the algorithm is robust in obtaining consistent results. Furthermore, feasible solutions are continuously found for all the test functions in 30 runs. These results reveal that ATMES has the substantial capability to deal with various kinds of COEAs.

We decided to empirically analyze the contribution of each tradeoff scheme proposed in each phase. The average number of generations each tradeoff scheme uses is summarized in Table III. As can be seen, for only six problems (g01, g05, g06, g07, g10, and g11), phase one can play its role, and for only four problems (g02, g08, g11, and g12), phase three can play its role. In addition, all problems will experience phase two. This phenomenon suggests that the tradeoff scheme in the second phase is the most important and dominant selection scheme for most problems during the evolutionary process. For problems g08 and g12, phase three is the main evolutionary stage. It is important to emphasize that these two problems share the following feature: the global optimum lying within the feasible region. Note that if the process of evolution does not undergo the first phase, the algorithm will be very efficient, since in this case the complexity of the algorithm is $O(\lambda \log(\lambda))$.

We have demonstrated the effectiveness of the hierarchical nondominated individual selection scheme proposed in Section III-A. We conduct our optimization run using a 2-D test function g06. In this run, the population can contain feasible solutions after seven iterations. Contour plots at generations 1, 3, and 7 are shown in Fig. 5. From Fig. 5, it is clear that the hierarchical nondominated individual selection scheme can motivate the population to approach the feasible region of the search space from different directions promptly.

C. Comparison With Stochastic Ranking (SR) and Simple Multimembered Evolutionary Strategy (SMES)

We compare our approach against two state-of-the-art approaches: the SR [4] and the SMES [5]. SR has been discussed

TABLE II
STATISTICAL RESULTS OBTAINED BY ATMES FOR 13 BENCHMARK TEST FUNCTIONS OVER 30 INDEPENDENT RUNS. A RESULT IN BOLDFACE INDICATES THAT THE GLOBAL OPTIMUM WAS REACHED

Fcn	Optimal	best	median	mean	worst	st. dev	average percentage
g01	-15.000	**-15.000**	**-15.000**	**-15.000**	**-15.000**	1.6E-14	75
g02	-0.803619	-0.803388	-0.792420	-0.790148	-0.756986	1.3E-02	74
g03	-1.000	**-1.000**	**-1.000**	**-1.000**	**-1.000**	5.9E-05	54
g04	-30665.539	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	7.4E-12	80
g05	5126.498	**5126.498**	5126.776	5127.648	5135.256	1.8E+00	49
g06	-6961.814	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	4.6E-12	62
g07	24.306	**24.306**	24.313	24.316	24.359	1.1E-02	57
g08	-0.095825	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	2.8E-17	100
g09	680.630	**680.630**	680.633	680.639	680.673	1.0E-02	99
g10	7049.248	7052.253	7215.357	7250.437	7560.224	1.2E+02	78
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	3.4E-04	73
g12	-1.000	**-1.000**	**-1.000**	**-1.000**	-0.994	1.0E-03	100
g13	0.053950	**0.053950**	0.053952	0.053959	0.053999	1.3E-05	70

TABLE III
THE AVERAGE NUMBER OF GENERATIONS EACH TRADEOFF SCHEME USES

Fcn	Average iteration number		
	phase one	phase two	phase three
g01	14	786	0
g02	0	783	17
g03	0	800	0
g04	0	800	0
g05	28	772	0
g06	6	794	0
g07	11	789	0
g08	0	18	782
g09	0	800	0
g10	18	782	0
g11	18	722	60
g12	0	168	632
g13	0	800	0

in Section II. Note that SR has been further improved by the same authors [27]. The improved SR (ISR) is one of the most competitive algorithms known to date. In addition to the adaptation mechanism of an ES and three simple comparison criteria, SMES also introduces three key components: a diversity mechanism that is intended to add some infeasible solutions into the next population, the combined recombination, and the reduction of the initial stepsize of the ES. It is important to emphasize that ES is adopted as the search algorithm by the three methods in comparison and the number of runs provided by them is 30.

As shown in Table IV, the performance of ATMES is compared in detail with SR and SMES using the selected performance metrics. For test functions g01, g03, g04, g08, and g11, the optimal solutions are consistently found by these three methods.

With respect to SR, our approach finds better "best" solutions in three test functions (g07, g10, and g13) and similar "best" results in nine test functions (g01, g03, g04, g05, g06, g08, g09, g11, and g12). A better "best" result is found by SR in test function g02. Also, our technique reaches better "mean" and "worst" results in seven test functions (g02, g05, g06, g07, g09, g10, and g13). Similar "mean" and "worst" results are found in five test functions (g01, g03, g04, g08, and g11). For test function g12, similar "mean" result is found. However, SR finds a "worst" result of higher quality.

Compared with SMES, our method finds better "best" solutions in four test functions (g05, g07, g09, and g13) and similar "best" results in seven test functions (g01, g03, g04, g06, g08, g11, and g12). Better "best" results are found by SMES in test functions g02 and g10. Our approach finds better "mean" and "worst" results in seven functions (g02, g05, g06, g07, g09, g10, and g13). It also provides similar "mean" and "worst" results in five test functions (g01, g03, g04, g08, and g11). Again, for test function g12, similar "mean" results are found, and SMES finds a better "worst" result.

As far as the computational cost (the number of FFEs) is concerned, SMES and ATMES have the minimum computational cost (240 000 FFEs) for all the test functions, while SR has a higher computational cost (350 000 FFEs) for all the test functions.

In summary, we can conclude that ATMES outperforms or performs similarly to SR and SMES in terms of the quality of the resulting solutions. In addition, SMES and ATMES are dominant in their efficiency. Another good property of ATMES is that it requires no problem dependent parameter. In contrast, SR is sensitive to its parameter p_f, and SMES needs to adapt the tolerance value of the equality constraints and the initial stepsize of the ES for different test functions. Furthermore, the parameter ε in ATMES is set to 5E-06, which is remarkably less than that used by SR and SMES, and makes the problems more difficult to solve.

D. Finding the Strength of ATM

One may be interested in the influence of ATM on the behavior of our algorithm, in other words, whether ATM really has the ability in balancing the objective function and the constraint violations as expected. Indeed, the direct consequence of the tradeoff between the objective function and the constraint

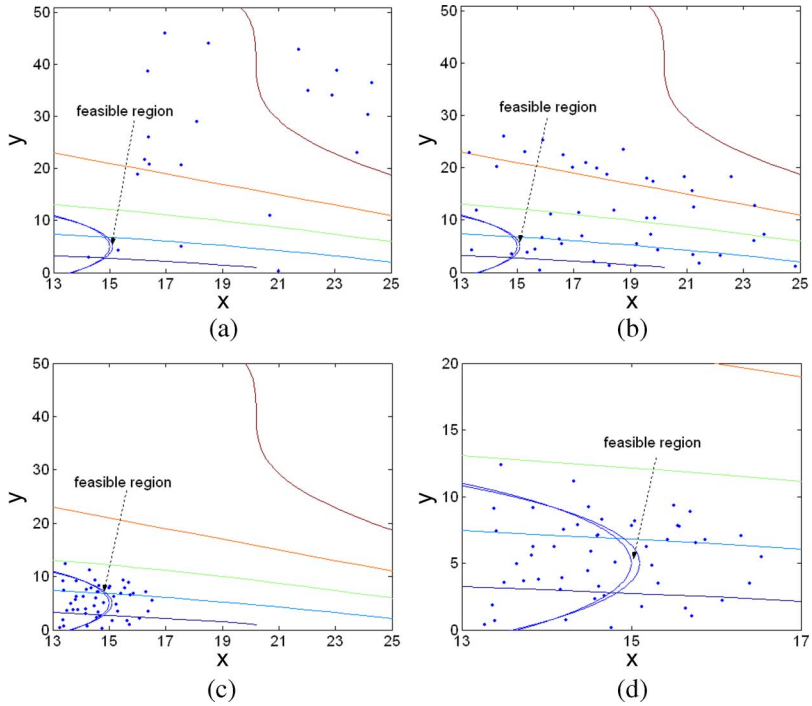


Fig. 5. Contour plots of problem 6. (a) Search space at generation 1. (b) Search space at generation 3. (c) Search space at generation 7. (d) A zooming of (c).

TABLE IV
COMPARING OUR ALGORITHM (INDICATED BY ATMES) WITH RESPECT TO SR [4] AND SMES [5] ON 13 BENCHMARK FUNCTIONS. A RESULT IN BOLDFACE INDICATES THE BEST RESULT AMONG THE THREE COMPARED ALGORITHMS OR THAT THE GLOBAL OPTIMUM WAS REACHED

Fcn	Optimal	Best Result			Mean Result			Worst Result		
		ATMES	SR	SMES	ATMES	SR	SMES	ATMES	SR	SMES
g01	-15.000	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**
g02	-0.803619	-0.803388	-0.803515	**-0.803601**	**-0.790148**	-0.781975	0.785238	**-0.756986**	-0.726288	-0.751322
g03	-1.000	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g04	-30665.539	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
g05	5126.498	**5126.498**	**5126.497**	5126.599	**5127.648**	5128.881	5174.492	**5135.256**	5142.472	5304.167
g06	-6961.814	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	-6875.940	-6961.284	**-6961.814**	-6350.262	-6952.482
g07	24.306	**24.306**	24.307	24.327	**24.316**	24.374	24.475	**24.359**	24.642	24.843
g08	-0.095825	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g09	680.630	**680.630**	**680.630**	680.632	**680.639**	680.656	680.643	**680.673**	680.763	680.719
g10	7049.248	7052.253	7054.316	**7051.903**	**7250.437**	7559.192	7253.047	**7560.224**	8835.655	7638.366
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	-1.000	**-1.000**	**-1.000**	**1.000**	**-1.000**	**-1.000**	**1.000**	-0.994	**-1.000**	**1.000**
g13	0.053950	**0.053950**	0.053957	0.053986	**0.053959**	0.067543	0.166385	**0.053999**	0.216915	0.468294

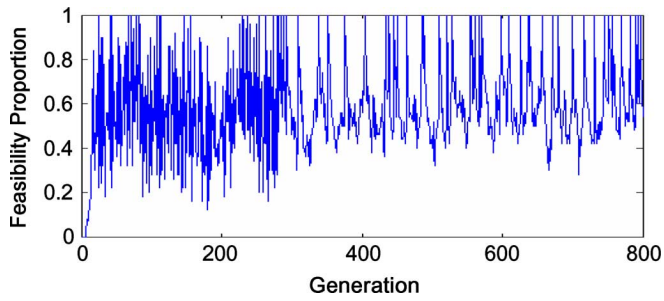


Fig. 6. Typical feasibility proportion versus generation for test function g06.

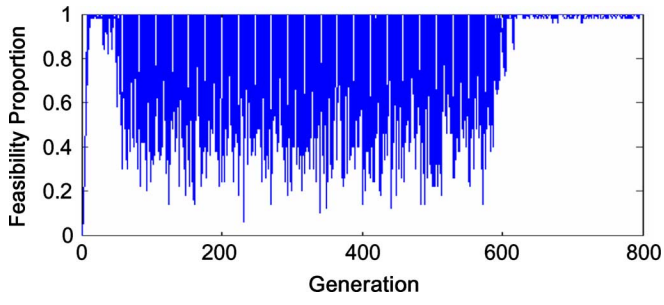


Fig. 7. Typical feasibility proportion versus generation for test function g09.

violations is the change of the feasibility proportion in the population during the evolution. Hence, we choose three test functions g06, g09, and g12, whose typical feasibility proportion is memorized at each generation by running once and shown in Figs. 6–8, to answer this question.

For test function g06 (see Fig. 6), at the initial stage, the proportion of feasible solutions in the population is 0 due to its relatively small feasible region. Then, the feasibility proportion quickly increases because of the hierarchical nondominated individual selection scheme applied. Furthermore, the feasibility

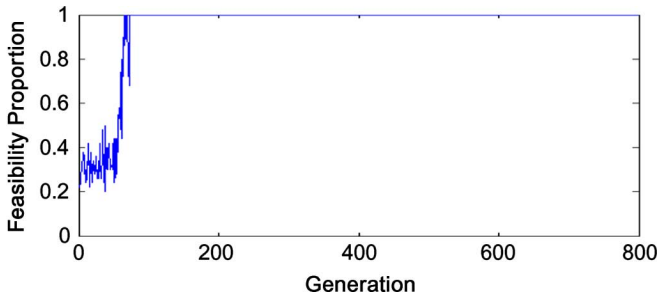


Fig. 8. Typical feasibility proportion versus generation for test function g12.

proportion bounds once it reaches a higher or lower value as the evolution proceeds. This phenomenon is reasonable, since if the feasibility proportion is too large or too small, ATM will play its role to adapt the feasibility proportion constantly by the conversion of fitness function. As shown in Figs. 7 and 8, the population can contain feasible solutions for problems g09 and g12 at the initial stage. Afterward, a relatively stable feasibility proportion will be attained gradually as the iteration increases. More specifically, for test function g09, the feasibility proportion will slightly bound but within a very small range at the later stage; for test function g12, the feasibility proportion converges to 1 in the end. This is because a large number of feasible solutions dominate infeasible solutions in both the objective function and the constraint violations. Such behaviors also suggest that, for some test functions, the population can reach a relatively steady feasibility proportion after the feasibility proportion is adjusted by ATM during some generations.

From the observations above, it can be seen that the feasibility proportion in the population either attains a relatively stable value (see Fig. 8) or region (see Fig. 7), or bounds within a range (see Fig. 6) over the course of the evolution. Furthermore, we can conclude that the proposed ATM turns out to have a remarkable potential in adaptively balancing the objective function and the constraint violations.

V. Discussion

In this section, the effect of algorithm parameters on the performance will be discussed through various experiments.

A. Effect of the Selection Scheme in Phase One

For examining the effect of the hierarchical nondominated individual selection scheme (HNISS) on the search ability of our ATMES, we perform computational experiments using various selection schemes. The original algorithm is denoted as HNISS. The algorithm selecting all nondominated individuals (instead of half) to the next population at each generation is denoted as HNISS+. The algorithm using an ordering scheme where only constraint violation is considered is denoted as HNISS-. Since phase one only plays its role in six problems (g01, g05, g06, g07, g10, and g11), we perform HNISS+ and HNISS- on these problems over 30 trials. Table V summarizes the experimental results.

As shown in Table V, with respect to problems g01, g05, and g10, HNISS+ cannot consistently find feasible solutions in the final populations, while HNISS+ performs similarly to HNISS for problems g06, g07 and g11. This may be because there is no search bias in this selection scheme and the search may wander deeply into an infeasible region with tempting objective function values and constraint violations. Thus, the population cannot approach the feasible region over time.

Additionally, the results of HNISS- for problems g05 and g10 are clearly dominated by those of HNISS, while the capability of HNISS- is similar to HNISS for the other four problems. As analyzed, this is because the diversity of the population in these cases is not good.

These results suggest that only selecting the first half of nondominated individuals in phase one is reasonable.

B. Effect of the Parameter φ in Phase Two

The parameter φ is effective for adjusting the tradeoff between the objective function and the constraint violations in phase two. For examining the effect of the parameter φ on the performance of our ATMES, we test phase two with five different φ: 0.0, 0.3, 0.5, 0.7, and 1.0. We summarize the mean of the objective function values in Table VI. It is worth noting that, when this parameter is specified as $\varphi = 0.0$, any feasible solution is preferred to any infeasible solution; when this parameter is specified as $\varphi = 1.0$, equation (7) in phase two only influences the infeasible individuals with objective function values less than that of the best feasible solution. Our original algorithm is referred to as "adaptation" in Table VI.

From Table VI, we observe a clear negative effect when the parameter φ is fixed to different values. The following is a summary of the comparison between the algorithms with fixed φ and the original algorithm.

1) $\varphi = 0.0$: In this case, compared with the original algorithm, premature convergence arises for problem g04. Moreover, we can see the performance deterioration for problems g02, g05, g07, g09, g10, g12, and g13. Similar results are obtained for problems g01, g03, g06, g08, and g11.
2) $\varphi = 0.3$: This case performs similarly to the algorithm with $\varphi = 0.0$. The only exception is that feasible solutions can only be found for 18 out of 30 trials for problem g10.
3) $\varphi = 0.5$: Although this case provides a better quality result for problem g02, it degrades its performance for problems g04, g07, g09, and g13. More importantly, feasible solutions cannot be found consistently for problems g05 and g10. Similar results are obtained for problems g01, g03, g06, g08, g11, and g12.
4) $\varphi = 0.7$: The performance of this case is similar to that of the algorithm with $\varphi = 0.5$. Note that it obtains a worse result than the original algorithm for problem g02.
5) $\varphi = 1.0$: This case is unable to consistently reach the feasible region for problems g05, g06, and g10. Particularly, for problem g10, the feasible solutions can only be found for 1 out of 30 trials. In addition, it provides results of a worse quality for problems g07, g09, and g13, it gives a better result for problem g02, and has similar capability for problems g01, g03, g04, g08, g11, and g12.

Based on the comparison above, we can see that the adaptive adjusting of the parameter φ has more stable and better performance than the fixed settings for this parameter.

TABLE V
EXPERIMENT RESULTS WITH 30 INDEPENDENT RUNS ON SIX BENCHMARK FUNCTIONS USING ALGORITHMS HNISS, HNISS+, AND HNISS-; (#) DENOTES THE NUMBER OF TRIALS IN WHICH FEASIBLE SOLUTIONS ARE FOUND IN THE FINAL POPULATIONS OVER 30 TRIALS

Fcn	method	best	median	mean	worst	st. dev
g01	HNISS	-15.000	-15.000	-15.000	-15.000	1.6E-14
	HNISS+			(0)		
	HNISS-	-15.000	-15.000	-15.000	-15.000	1.6E-14
g05	HNISS	5126.498	5126.776	5127.648	5135.256	1.8E+00
	HNISS+			(0)		
	HNISS-	5126.509	5129.033	5158.713	5724.586	1.1E+02
g06	HNISS	-6961.814	-6961.814	-6961.814	-6961.814	4.6E-12
	HNISS+	-6961.814	-6961.814	-6961.814	-6961.814	4.6E-12
	HNISS-	-6961.814	-6961.814	-6961.814	-6961.814	4.6E-12
g07	HNISS	24.306	24.313	24.316	24.359	1.1E-02
	HNISS+	24.307	24.310	24.316	24.354	1.2E-02
	HNISS-	24.306	24.311	24.314	24.332	7.8E-03
g10	HNISS	7052.253	7215.357	7250.437	7560.224	1.2E+02
	HNISS+			(18)		
	HNISS-	7065.868	7252.623	7280.931	7758.223	1.3E+02
g11	HNISS	0.75	0.75	0.75	0.75	3.4E-04
	HNISS+	0.75	0.75	0.75	0.75	7.3E-04
	HNISS-	0.75	0.75	0.75	0.75	7.5E-04

TABLE VI
EXPERIMENTAL RESULTS ON 13 BENCHMARK FUNCTIONS WITH VARYING φ; 30 INDEPENDENT RUNS ARE PERFORMED; A RESULT IN BOLDFACE INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) IS REACHED; (#) DENOTES THE NUMBER OF TRIALS IN WHICH FEASIBLE SOLUTIONS ARE FOUND IN THE FINAL POPULATIONS OVER 30 TRIALS

Fcn	0.0	0.3	0.5	0.7	1.0	adaptation
g01	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**	**-15.000**
g02	-0.788658	-0.785795	**-0.792233**	-0.785698	-0.790508	-0.790148
g03	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g04	-30665.517	-30665.421	-30665.525	-30665.534	**-30665.539**	**-30665.539**
g05	5127.915	5129.019	(29)	(29)	(23)	**5127.648**
g06	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	(21)	**-6961.814**
g07	24.373	24.372	24.366	24.365	24.339	**24.316**
g08	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g09	680.647	680.662	680.642	680.654	680.643	**680.639**
g10	7384.358	(18)	(27)	(12)	(1)	**7250.437**
g11	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	-0.965	-0.998	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g13	0.053964	0.053965	0.053967	**0.053959**	0.053960	**0.053959**

C. Effect of the Initial Stepsize Reduction of the ES

Since the number of iterations used in this paper is nearly half of that used in SR, the initial stepsize of the ES in this paper should be readjusted to meet competitive results. Table VII summarizes the mean of the objective function values in the case of the initial stepsize reduction (i.e., $\breve{\sigma}$) being set to 0.2, 0.4, 0.6, 0.8, and 1.0.

In case of $\breve{\sigma} = 0.2$, while the algorithm can obtain the best result for problem g10, the results for problems g01 and g02 are much worse than other results. Furthermore, for problem g03, the algorithm is unable to reach the feasible region consistently. In the case of $\breve{\sigma} = 0.4$, the results for problems g02 and g10 are worse than those of $\breve{\sigma} = 0.6, 0.8$ and 1.0. In the case of $\breve{\sigma} = 1.0$, premature convergence tends to occur for problem g12. On the average, the difference in the results is marginal when $\breve{\sigma} = 0.6$ and 0.8.

Based on the analyses above, we conclude that a value between 0.6 and 0.8 is suitable for the parameter $\breve{\sigma}$.

D. Effect of the Tolerance Value With Equality Constraints

In order to illustrate the effect of the tolerance value with equality constraints on the performance of our ATMES, a set of experiments have been performed on problems g03, g05, g11, and g13.[5] Table VIII summarizes the mean of the objective function values in the case of the parameter $\breve{\varepsilon}$ being set to 1.0101, 1.0129, 1.0159, 1.0168, and 1.0188, where the corresponding tolerance values when the procedure halts are 1E-03, 1E-04, 1E-05, 5E-06, and 1E-06 (i.e., $\varepsilon = 1\mathrm{E}-03$, 1E-04, 1E-05, 5E-06, and 1E-06).

From Table VIII, we can argue that there is no significant effect when decreasing the parameter $\breve{\varepsilon}$ from 1.0168 to 1.0101. Nevertheless, a lower value of $\breve{\varepsilon}$ also indicates a higher value of ε, thus, for some problems the "best" results obtained may be better than the "known" optima. However, this type of behaviors does not mean the "new" optima are really found. Based on our observations, the "best" results provided by $\varepsilon = 1\mathrm{E}-03$,

[5]Note that only these problems have equality constraints.

TABLE VII
EXPERIMENTAL RESULTS ON 13 BENCHMARK FUNCTIONS WITH VARYING $\breve{\sigma}$; 30 INDEPENDENT RUNS ARE PERFORMED; A RESULT IN BOLDFACE INDICATES A BETTER RESULT OR THAT THE GLOBAL OPTIMUM (OR BEST KNOWN SOLUTION) IS REACHED; (#) DENOTES THE NUMBER OF TRIALS IN WHICH FEASIBLE SOLUTIONS ARE FOUND IN THE FINAL POPULATIONS OVER 30 TRIALS

Fcn	0.2	0.4	0.6	0.8	1.0
g01	-14.414	**-15.000**	**-15.000**	**-15.000**	**-15.000**
g02	-0.577665	-0.763587	-0.787625	**-0.790148**	-0.789367
g03	(28)	**-1.000**	**-1.000**	**-1.000**	**-1.000**
g04	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
g05	5128.224	5127.706	5127.765	**5127.648**	5127.840
g06	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**
g07	24.322	**24.313**	24.315	24.316	24.315
g08	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**	**-0.095825**
g09	680.646	680.644	**680.638**	680.639	680.641
g10	**7168.140**	7273.581	7255.247	7250.437	7268.675
g11	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**
g12	**-1.000**	**-1.000**	**-1.000**	**-1.000**	-0.988
g13	0.053959	**0.053958**	**0.053958**	0.053959	0.053962

TABLE VIII
EXPERIMENTAL RESULTS ON FOUR BENCHMARK FUNCTIONS (g03, g05, g11, AND, g13) WITH VARYING TOLERANCE VALUE WITH EQUALITY CONSTRAINTS; 30 INDEPENDENT RUNS ARE PERFORMED; (#) DENOTES THE NUMBER OF TRIALS IN WHICH FEASIBLE SOLUTIONS ARE FOUND IN THE FINAL POPULATIONS OVER 30 TRIALS

Fcn	$\breve{\varepsilon}$ =1.0101 ε =1E-03	$\breve{\varepsilon}$ =1.0129 ε =1E-04	$\breve{\varepsilon}$ =1.0159 ε =1E-05	$\breve{\varepsilon}$ =1.0168 ε =5E-06	$\breve{\varepsilon}$ =1.0188 ε =1E-06
g03	-1.004	-1.000	-1.000	-1.000	(23)
g05	5127.248	5127.144	5127.674	5127.648	5127.826
g11	0.75	0.75	0.75	0.75	0.75
g13	0.053877	0.053947	0.053957	0.053959	0.053956

1E-04, and 1E-05 are better than the "known" optima for the four tested problems in different degrees. In these cases, the "best" results provided by $\varepsilon = 5\mathrm{E}-06$ are very close to the "known" optimal solutions. On the other hand, the performance degradation occurs for problem g03 when $\varepsilon = 1\mathrm{E}-06$, where feasible solutions can be found in the final populations for 23 out of the 30 runs. This negative impact on performance occurs because the tolerance value with equality constraints is too small to be satisfied for the algorithm.

The above observations validate that the setting of $\breve{\varepsilon} = 1.0168$ is an appropriate choice for equality constraints.

VI. CONCLUSION

We have analyzed some existing methods from the tradeoff point-of-view. Based on our analyses, the adaptive tradeoff between the objective function and the constraint violations has often been ignored by previous work. In this paper, we present an ATM which adopts a variety of tradeoff schemes in different search stages. Furthermore, we combine ATM with a simple ES and obtain a generic COEA, namely, ATMES. The experimental results suggest that ATMES gives results that are better or comparable to those of two other state-of-the-art techniques (SR and SMES). Apart from finding very competitive results, the proposed algorithm also finds feasible solutions in every run.

ISR [27], CW [28], and HCOEA [29] are the most competitive algorithms in constrained optimization so far. The strength of ISR is that it can reach very competitive results with no need to adapt parameters for different problems. However, based on the experimental results, it seems to be difficult for ISR to improve the performance on multimodal problems, such as g02. The highlight of CW is that it can reach competitive results without the transformation of equality constraints into inequality constraints. With respect to HCOEA, the strength is its efficiency. In general, CW and HCOEA are quite effective when solving COPs with equality constraints. However, both of them have a problem-dependent parameter for crossover operator, which limits the real-world application of the algorithms. The main advantages of ATMES are its simplicity and adaptation; however, compared with ISR, CW, and HCOEA, ATMES leaves plenty of room for improvement. So, the future work of this study includes the application of ATM to other types of search engines, such as genetic algorithms, differential evolution, and particle swarm optimizer.

ACKNOWLEDGMENT

The authors would like to thank Dr. T. P. Runarsson and Prof. X. Yao to provide the source code of the SR-based method. The authors sincerely thank the five anonymous reviewers and the anonymous associate editor for their valuable and constructive comments and suggestions. They also gratefully acknowledge Dr. Q. Cai and Dr. J. Cai for improving the presentation of this paper.

REFERENCES

[1] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithm for constrained parameter optimization problems," *Evol. Comput.*, vol. 4, no. 1, pp. 1–32, Feb. 1996.

[2] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Meth. Appl. Mech. Eng.*, vol. 191, no. 11–12, pp. 1245–1287, Jan. 2002.

[3] Z. Michalewicz, K. Deb, M. Schmidt, and T. Stidsen, "Test-case generator for constrained parameter optimization techniques," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 197–215, Jun. 2000.

[4] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Jun. 2000.

[5] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 1–17, Feb. 2005.
[6] K. Rasheed, "An adaptive penalty approach for constrained genetic algorithm optimization," in *Proc. 3rd Annu. Conf. Genetic Programming (GP-98)/Symp. Genetic Algorithms (SGA-98)*, 1998, pp. 584–590.
[7] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 7, no. 5, pp. 445–455, Oct. 2003.
[8] J. A. Wright and R. Farmani, "Genetic algorithm: A fitness formulation for constrained minimization," in *Proc. Genetic Evol. Comput. Conf.*, San Francisco, CA, 2001, pp. 725–732.
[9] D. Powell and M. M. Skolnick, "Using genetic algorithm in engineering design optimization with nonlinear constraint," in *Proc. 5th Int. Conf. Genetic Algorithms*, S. Forrest, Ed., 1993, pp. 424–431.
[10] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Meth. Appl. Mech. Eng.*, vol. 18, pp. 311–338, 2000.
[11] 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, no. 5, pp. 437–451, Oct. 2005.
[12] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms and Their Applications*, J. J. Grefenstette, Ed., Hillsdale, NJ, 1985, pp. 93–100.
[13] P. D. Surry and N. J. Radcliffe, "The COMOGA method: Constrained optimization by multiobjective genetic algorithm," *Control Cybern.*, vol. 26, no. 3, pp. 391–412, 1997.
[14] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 4, pp. 424–435, Aug. 2005.
[15] K. Deb, A. Pratab, S. Agrawal, and T. Meyarivan, "A fast and elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA?," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 182–197, Apr. 2002.
[16] Y. Zhou, Y. Li, J. He, and L. Kang, "Multiobjective and MGG evolutionary algorithm for constrained optimization," in *Proc. Congr. Evol. Comput. 2003 (CEC'2003)*, 2003, pp. 1–5.
[17] C. A. C. Coello, "Treating constraints as objectives for single-objective evolutionary optimization," *Eng. Opt.*, vol. 32, no. 3, pp. 275–308, 2000.
[18] C. A. C. Coello, "Constraint handling using an evolutionary multiobjective optimization technique," *Civil Eng. Environm. Syst.*, vol. 17, pp. 319–346, 2000.
[19] 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. A, Syst. Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1999.
[20] C. A. C. Coello and E. Mezura-Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Adv. Eng. Inf.*, vol. 16, no. 3, pp. 193–203, 2002.
[21] J. Horn, N. Nafpliotis, and D. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc 1st IEEE Conf. Evol. Comput.*, 1994, pp. 82–87.
[22] A. H. Aguirre, S. B. Rionda, C. A. C. Coello, G. L. Lizáraga, and E. M. Montes, "Handling constraints using multiobjective optimization concepts," *Int. J. Numerical Methods Eng.*, vol. 59, no. 15, pp. 1989–2017, Apr. 2004.
[23] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolutionary strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.
[24] T. Ray and K. M. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 386–396, Aug. 2003.
[25] S. B. Hamida and M. Schoenauer, "ASCHEA: New results using adaptive segregational constraint handling," in *Proc. Congr. Evol. Comput. 2002 (CEC'2002)*, 2002, pp. 82–87.
[26] E. Mezura-Montes and C. A. C. Coello, "Adding a diversity mechanism to a simple evolution strategy to solve constrained optimization problems," in *Proc. Congr. Evol. Comput. 2003 (CEC'2003)*, Dec. 2003, vol. 1, pp. 6–13.
[27] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 233–243, May 2005.
[28] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 658–675, Dec. 2006.
[29] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 560–575, Jun. 2007.

Yong Wang was born in Hubei, China, on December 26, 1980. He received the M.S. degree in pattern recognition and intelligent systems from Central South University (CSU), Changsha, China, in 2006. He is currently working towards the Ph.D. degree at CSU.

His current research interests include the theory of evolutionary computation, constrained optimization, multiobjective optimization and their applications in multiple mobile robots.

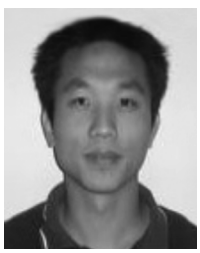
Zixing Cai (SM'98) received the Diploma degree from the Department of Electrical Engineering from Jiao Tong University, Xi'an, China, in 1962.

He has been teaching and doing research at the College of Information Science and Engineering, Central South University (CSU), Changsha, Hunan, China, since 1962. From May 1983 to December of 1983, he visited the Center of Robotics, Department of Electrical Engineering and Computer Science, University of Nevada, Reno. Then, he visited the Advanced Automation Research Laboratory, School of Electrical Engineering, Purdue University, West Lafayette, IN, from December 1983 to June 1985. From October 1988 to September 1990, he was a Senior Research Scientist with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Science, and the National Laboratory of Machine Perception, Center of Information, Beijing University. From September 1992 to March 1993, he visited the Center for Intelligent Robotic Systems for Space Exploration, Department of Electrical, Computer and System Engineering, Rensselaer Polytechnic Institute, Troy, NY, as a Visiting Professor. From April 2004 to July 2004, he visited the Institute of Informatics and Automation, Russia Academy of Sciences. Since February 1989, he has become an Expert of the United Nations granted by UNIDO. Over 500 papers and 25 books/textbooks have been published. His research interests include intelligent systems, artificial intelligence, intelligent computation, and robotics.

Prof. Cai received over 30 State, Province, University Awards in science, technology and teaching. One of the newest prizes is the State Eminent Professor Prize of China.

Yuren Zhou received the B.Sc. degree in the mathematics from Peking University, Beijing, China, in 1988, the M.Sc. degree in mathematics and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 1991 and 2003, respectively.

He is currently an Associate Professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests are focused on evolutionary computation and data mining.



Wei Zeng was born in Hubei, China, on November 30, 1981. He is working towards the M.S. degree at the College of Information Science and Engineering, Central South University (CSU), Changsha, Hunan, China.

His current research interests include constrained evolutionary optimization and multiobjective optimization.