Simple Feasibility Rules and Differential Evolution for Constrained Optimization

Efrén Mezura-Montes†, Carlos A. Coello Coello†and Edy I.Tun-Morales‡

CINVESTAV-IPN
†Evolutionary Computation Group (EVOCINV)
Departamento de Ingeniería Eléctrica
Sección de Computación
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
emezura@computacion.cs.cinvestav.mx
ccoello@cs.cinvestav.mx
‡Instituto Tecnológico de Villahermosa
Carretera Villahermosa-Frontera Km. 3.5 Cd. Industrial
Villahermosa Tabasco
eltedy@hotmail.com

Abstract. In this paper, we propose a differential evolution algorithm to solve constrained optimization problems. Our approach uses three simple selection criteria based on feasibility to guide the search to the feasible region. The proposed approach does not require any extra parameters other than those normally adopted by the Differential Evolution algorithm. The present approach was validated using test functions from a well-known benchmark commonly adopted to validate constraint-handling techniques used with evolutionary algorithms. The results obtained by the proposed approach are very competitive with respect to other constraint-handling techniques that are representative of the state-of-the-art in the area.

1 Introduction

Evolutionary Algorithms (EAs) are heuristics that have been successfully applied in a wide set of areas [1, 2], both in single and in multiobjective optimization. However, EAs lack a mechanism able to bias efficiently the search towards the feasible region in constrained search spaces. This has triggered a considerable amount of research and a wide variety of approaches have been suggested in the last few years to incorporate constraints into the fitness function of an evolutionary algorithm [3, 4].

The most common approach adopted to deal with constrained search spaces is the use of penalty functions. When using a penalty function, the amount of constraint violation is used to punish or "penalize" an infeasible solution so that feasible solutions are favored by the selection process. Despite the popularity of penalty functions, they have several drawbacks from which the main one is that they require a careful fine tuning of the penalty factors that accurately estimates the degree of penalization to be applied as to approach efficiently the feasible region [5, 3].

Differential Evolution (DE) is a relatively new EA proposed by Price and Storn [6]. The algorithm is based on the use of a special crossover-mutation operator, based on the linear combination of three different individuals and one subject-to-replacement parent. The selection process is performed via deterministic tournament selection between the parent and the child created by it. However, as any other EA, DE lacks a mechanism to deal with constrained search spaces.

The constraint-handling approach proposed in this paper relies on three simple selection criteria based on feasibility to bias the search towards the feasible region. We have used the same approach implemented on different types of Evolution Strategies in which the results were very promising [7, 8]. The main motivation of this work was to analyze if the use of the selection criteria that we successfully adopted in evolution strategies would also work with differential evolution. This is an important issue to us, because it has been hypothesized in the past that evolution strategies are a very powerful search engine for constrained optimization when dealing with real numbers [9]. However, no such studies exist for differential evolution nor other related heuristics that operate on real numbers (as evolution strategies). We thus believe that the search power of other heuristics such as differential evolution has been underestimated and therefore our interest in analyzing such search power.

The paper is organized as follows: In Section 2, the problem of our interest is stated. In Section 3 we describe the previous work related with the current algorithm. A detailed description of our approach is provided in Section 4. The experiments performed and the results obtained are shown in Section 5 and in Section 6 we discuss them. Finally, in Section 7 we establish some conclusions and we define our future paths of research.

2 Statement of the Problem

We are interested in the general nonlinear programming problem in which we want to:Find $\boldsymbol{x}$ which optimizes $f(\boldsymbol{x})$ subject to: $g_i(\boldsymbol{x}) \leq 0,\ \ i = 1, \ldots, n$ $h_j(\boldsymbol{x}) = 0,\ \ j = 1, \ldots, p$ where $\boldsymbol{x}$ is the vector of solutions $\boldsymbol{x} = [x_1, x_2, \ldots, x_r]^T$, n is the number of inequality constraints and p is the number of equality constraints (in both cases, constraints could be linear or nonlinear). If we denote with $\mathcal{F}$ to the feasible region and with $\mathcal{S}$ to the whole search space, then it should be clear that $\mathcal{F} \subseteq \mathcal{S}$. For an inequality constraint that satisfies $g_i(\boldsymbol{x}) = 0$, we will say that is active at $\boldsymbol{x}$. All equality constraints h_j (regardless of the value of $\boldsymbol{x}$ used) are considered active at all points of $\mathcal{F}$.

3 Previous Work

DE is a population-based evolutionary algorithm with an special recombination operator that performs a linear combination of a number of individuals (normally three) and one parent (which is subject to be replaced) to create one child. The selection is deterministic between the parent and the child. The best of them remain in the next population. DE shares similarities with traditional EAs. However it does not use binary encoding as a simple genetic algorithm [2] and it does not use a probability density function to self-adapt its parameters as an Evolution Strategy [10]. The main differential evolution algorithm [6] is presented in Figure 1.

Begin
  G=0
  Create a random initial population $\boldsymbol{x}_G^i \; \forall i, i = 1, \ldots, NP$
  Evaluate $f(\boldsymbol{x}_G^i) \; \forall i, i = 1, \ldots, NP$
  For G=1 to MAX_GENERATIONS **Do**
    For i=1 to NP **Do**
      Select randomly $r_1 \neq r_2 \neq r_3$:
      $j_{rand} =$ randint$(1, D)$
      For j=1 to D **Do**
        If ($rand_j[0, 1) < CR$ **or** $j = j_{rand}$) **Then**
          $u_{j,G+1}^i = x_{j,G}^{r_3} + F(x_{j,G}^{r_1} - x_{j,G}^{r_2})$
        Else
          $u_{j,G+1}^i = x_{j,G}^i$
        End If
      End For
      If ($f(\boldsymbol{u}_{G+1}^i) \leq f(\boldsymbol{x}_G^i)$) **Then**
        $\boldsymbol{x}_{G+1}^i = \boldsymbol{u}_{G+1}^i$
      Else
        $\boldsymbol{x}_{G+1}^i = \boldsymbol{x}_G^i$
      End If
    End For
    $G = G + 1$
  End For
End

Fig. 1. DE algorithm. randint(min,max) is a function that returns an integer number between min and max. rand$[0, 1)$ is a function that returns a real number between 0 and 1. Both are based on a uniform probability distribution. “NP”, “MAX_GENERATIONS”, “CR” and “F” are user-defined parameters.

The use of tournament selection based on feasibility rules has been explored by other authors. Jiménez and Verdegay [11] proposed an approach similar to a min-max formulation used in multiobjective optimization combined with tournament selection. The rules used by them are similar to those adopted in this work. However, Jiménez and Verdegay’s approach lacks an explicit mechanism to avoid the premature convergence produced by the random sampling of the feasible region because their approach is guided by the first feasible solution found. Deb [12] used the same tournament rules previously indicated in his approach. However, Deb proposed to use niching as a diversity mechanism, which introduces some extra computational time (niching has time-complexity $O(N^2)$). In Deb’s approach, feasible solutions are always considered better than infeasible ones. This contradicts the idea of allowing infeasible individuals to remain in the population. Therefore, this approach will have difficulties in problems in which the global optimum lies on the boundary between the feasible and the infeasible regions. Coello & Mezura [13] used tournament selection based on feasibility rules. They also adopted nondominance checking using a sample of the population (as the multiobjective optimization approach called NPGA [14]). They adopted a user-defined parameter S_r, to control the diversity in the population. This approach provided good results in some well-known engineering problems and in some benchmark problems, but presented problems when facing high dimensionality [13].

Some previous approaches have been proposed to solve constrained optimization problems using DE. Storn [15] proposed an adaptive mechanism that relaxes the constraints of the problem in order to make all the initial solutions feasible. This pseudo-feasible region is shrunk each generation until it matches the real feasible region. Also, Storn [15] proposed to use an aging concept in order to avoid that a solution remains in the population too many generations. Furthermore, he modified the original DE algorithm because when a child is created and it is not better than the parent subject-to-replace, another child is created. The process is repeated NT times. If the parent is still better, the parent remains in the population. Both, the aging parameter and NT are defined by the user. Storn [15] used a modified "DE/rand/1/bin" version. The approach showed a good performance in problems with only inequality constraints but presented problems when dealing with equality constraints. Moreover, only two test functions (out of seven used to test the approach) are included in the well-known benchmark for constrained optimization proposed by Koziel & Michalewicz [16] and enriched by Runarsson & Yao [9]. The main drawback of the approach is that it adds two user-defined parameters and that the NT parameter can cause an increase in the number of evaluations of the objective function without any user control.

Lampinen & Zelinka [17] used DE to solve engineering design problems. They opted to handle constraints using a static penalty function approach that they called "Soft -constraint". The authors tested their approach using three well-known engineering design problems [17]. They compared their results with respect to several classical techniques and with respect to some heuristic methods. The main drawback of the approach is the careful tuning required for the penalty factors which is in fact mentioned by the authors in their article. The last two methods discussed also lack of a mechanism to maintain diversity (to have both, feasible and infeasible solutions in the population during all the evolutionary process), which is one of the most important aspects to consider when designing a competitive constraint-handling approach [8].

4 Our approach

The design of our approach is based on the idea of preserving the main DE algorithm and just adding a simple mechanism, which has been found to be successful with other EAs. Moreover, our constraint-handling approach does not add any extra parameter defined by the user (other than those required by the original DE algorithm).

The modifications made to the original DE are the following:

1. The simple mechanism to deal with constraints are three simple selection criteria which guide the algorithm to the feasible region of the search space:
   - Between 2 feasible solutions, the one with the highest fitness value wins.
   - If one solution is feasible and the other one is infeasible, the feasible solution wins.
   - If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

These criteria are applied when the child is compared against the parent subject to be replaced.

2. In order to accelerate the convergence process, when a child replaces its parent, it is copied into the new generation but it is also copied into the current generation. The goal of this change is to allow the new child, which is a new and better solution, to be selected among the three solutions (r_1, r_2 or r_3) and contribute to create better solutions. In this way, a promising solution does not need to wait for the next generation to share its genetic code.
3. When a new decision variable of the child is created and it is out of the limits established (lower and upper) by an amount, this amount is subtracted or added to the limit violated to shift the value inside the limits. If the shifted value is now violating the other limit (which may occur), as a last option, a random value inside the limits is generated.

Our proposed version of the DE algorithm, called CHDE (Constraint Handling Differential Evolution) is shown in Figure 2.

Begin
  G=0
  Create a random initial population $\boldsymbol{x}_G^i \; \forall i, i = 1, \ldots, NP$
  Evaluate $f(\boldsymbol{x}_G^i) \; \forall i, i = 1, \ldots, NP$
  For G=1 to MAX_GENERATIONS **Do**
    For i=1 to NP **Do**
      Select randomly $r_1 \neq r_2 \neq r_3$:
      $j_{rand} =$ randint$(1, D)$
      For j=1 to D **Do**
        If $(rand_j[0, 1) < CR$ **or** $j = j_{rand})$ **Then**
          $u_{j,G+1}^i = x_{j,G}^{r_3} + F(x_{j,G}^{r_1} - x_{j,G}^{r_2})$
        Else
          $u_{j,G+1}^i = x_{j,G}^i$
        End If
      End For
$\Longrightarrow$ **If** ($\boldsymbol{u}_{G+1}^i$ is better than $\boldsymbol{x}_G^i$ (based on the three selection criteria)) **Then**
        $\boldsymbol{x}_{G+1}^i = \boldsymbol{u}_{G+1}^i$
$\Longrightarrow$ $\boldsymbol{x}_G^i = \boldsymbol{u}_{G+1}^i$
      Else
        $\boldsymbol{x}_{G+1}^i = \boldsymbol{x}_G^i$
$\Longrightarrow$ $\boldsymbol{x}_G^i = \boldsymbol{x}_G^i$
      End If
    End For
    $G = G + 1$
  End For
End

Fig. 2. CHDE algorithm. The modified steps are marked with an arrow. randint(min,max) is a function that returns an integer number between min and max. rand$[0, 1)$ is a function that returns a real number between 0 and 1. Both are based on a uniform probability distribution. "NP", "MAX_GENERATIONS", "CR" and "F" are user-defined parameters

5 Experiments and Results

To evaluate the performance of the proposed approach we used the 13 test functions described in [9]. The test functions chosen contain characteristics that are representative of what can be considered "difficult" global optimization problems for an evolutionary algorithm. Their expressions can be found in [9]

To get a measure of the difficulty of solving each of these problems, a ρ metric (as suggested by Koziel and Michalewicz [16]) was computed using the following expression: $\rho = |F|/|S|$, where $|F|$ is the number of feasible solutions and $|S|$ is the total number of solutions randomly generated. In this work, $S = 1,000,000$ random solutions.

Problem	n	Type of function	ρ	LI	NI	LE	NE
g01	13	quadratic	0.0003%	9	0	0	0
g02	20	nonlinear	99.9973%	2	0	0	0
g03	10	nonlinear	0.0026%	0	0	0	1
g04	5	quadratic	27.0079%	4	2	0	0
g05	4	nonlinear	0.0000%	2	0	0	3
g06	2	nonlinear	0.0057%	0	2	0	0
g07	10	quadratic	0.0000%	3	5	0	0
g08	2	nonlinear	0.8581%	0	2	0	0
g09	7	nonlinear	0.5199%	0	4	0	0
g10	8	linear	0.0020%	6	0	0	0
g11	2	quadratic	0.0973%	0	0	0	1
g12	3	quadratic	4.7697%	0	9^3	0	0
g13	5	nonlinear	0.0000%	0	0	1	2

Table 1. Values of ρ for the 13 test problems chosen.

The different values of ρ for each of the functions chosen are shown in Table 1, where n is the number of decision variables, LI is the number of linear inequalities, NI the number of nonlinear inequalities, LE is the number of linear equalities and NE is the number of nonlinear equalities.

We performed 30 independent runs for each test function. Equality constraints were transformed into inequalities using a tolerance value of 0.0001 (except for problems g03, g11 and g13 where the tolerance was 0.001). The parameters used for the CHDE are the following: $NP = 60$, $MAX\_GENERATIONS = 5,800$. To ensure that there is no sensitivity to "F" and "CR" parameters, F was generated randomly (using a uniform distribution) per run between $[0.3, 0.9]$ and CR was also randomly generated between $[0.8, 1.0]$. The intervals for both parameters were defined empirically.

The results obtained with the CHDE are presented in Table 2. A comparison of the performance of CHDE with respect to three techniques that are representative of the state-of-the-art in the area: the Homomorphous maps [16], Stochastic Ranking [9] and the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA) [18] are presented in Tables 3, 4 and 5, respectively.

	Statistical Results of the CHDE Algorithm					
Problem	Optimal	Best	Mean	Median	Worst	St. Dev.
g01	−15	−15.000	−14.792134	−15.000	−12.743044	0.401
g02	0.803619	0.803619	0.746236	0.800445	0.302179	0.081
g03	1	1.00	0.640326	0.702939	0.029601	0.239
g04	−30665.539	−30665.539	−30592.154435	−30665.539	−29986.214382	108.779
g05	5126.498	5126.496714	5218.729114	5231.557639	5502.410392	76.422
g06	−6961.814	−6961.814	−6367.575424	−6961.814	−2236.950336	770.803
g07	24.306	24.306	104.599221	24.482980	1120.541494	176.761
g08	0.095825	0.095825	0.091292	0.095825	0.027188	0.012
g09	680.63	680.6300	692.472322	680.639178	839.782911	23.575
g10	7049.25	7049.248021	8442.656946	7137.415303	15580.370333	2186.49
g11	0.75	0.749	0.761823	0.749	0.870984	0.020
g12	1	1	1	1	1	0
g13	0.053950	0.053866	0.747227	0.980831	2.259875	0.313

Table 2. Statistical results obtained by the CHDE for the 13 test functions with 30 independent runs.

		Best Result		Mean Result		Worst Result	
Problem	Optimal	CHDE	HM	CHDE	HM	CHDE	HM
g01	−15	−15.000	−14.7886	−14.792134	−14.7082	−12.743044	−14.6154
g02	0.803619	0.803619	0.79953	0.746236	0.79671	0.302179	0.79119
g03	1	1.00	0.9997	0.640326	0.9989	0.029601	0.9978
g04	−30665.539	−30665.539	−30664.5	−30592.154435	−30655.3	−29986.214382	−30645.9
g05	5126.498	5126.496714	−	5218.729114	−	5502.410392	−
g06	−6961.814	−6961.814	−6952.1	−6367.575424	−6342.6	−2236.950336	−5473.9
g07	24.306	24.306	24.620	104.599221	24.826	1120.541494	25.069
g08	0.095825	0.095825	0.0958250	0.091292	0.0891568	0.027188	0.0291438
g09	680.63	680.6300	680.91	692.472322	681.16	839.782911	683.18
g10	7049.25	7049.248021	7147.9	8442.656946	8163.6	15580.370333	9659.3
g11	0.75	0.749	0.75	0.761823	0.75	0.870984	0.75
g12	1	1	0.999999857	1	0.999134613	1	0.991950498
g13	0.053950	0.053866	NA	0.747227	NA	2.259875	NA

Table 3. Comparison of our approach (CHDE) with respect to the Homomorphous Maps (HM) NA = Not Available.

		Best Result		Mean Result		Worst Result	
Problem	Optimal	CHDE	SR	CHDE	SR	CHDE	SR
g01	−15	−15.000	−15.000	−14.792134	−15.000	−12.743044	−15.000
g02	0.803619	0.803619	0.803515	0.746236	0.781975	0.302179	0.726288
g03	1	1.00	1.000	0.640326	1.000	0.029601	1.000
g04	−30665.539	−30665.539	−30665.539	−30592.154435	−30665.539	−29986.214382	−30665.539
g05	5126.498	5126.496714	5126.497	5218.729114	5128.881	5502.410392	5142.472
g06	−6961.814	−6961.814	−6961.814	−6367.575424	−6875.940	−2236.950336	−6350.262
g07	24.306	24.306	24.307	104.599221	24.374	1120.541494	24.642
g08	0.095825	0.095825	0.095825	0.091292	0.095825	0.027188	0.095825
g09	680.63	680.6300	680.630	692.472322	680.656	839.782911	680.763
g10	7049.25	7049.248021	7054.316	8442.656946	7559.192	15580.370333	8835.655
g11	0.75	0.749	0.750	0.761823	0.750	0.870984	0.750
g12	1	1	1	1	1	1	1
g13	0.053950	0.053866	0.053957	0.747227	0.057006	2.259875	0.216915

Table 4. Comparison of our approach (CHDE) with respect to the Stochastic Ranking (SR)

6 Discussion of Results

As can be seen in Table 2, CHDE could reach the global optimum in the 13 test problems. The apparent improvement to the optimum solutions (or the best-known solutions) for problems g03, g05, g11 and g13 is due to the tolerance value adopted for the equality constraints. However, the statistical measures suggest that the proposed approach presents premature convergence in some cases. This seems to be originated by the high selection pressure provided by the deterministic selection. It also causes that infeasible solutions close to the boundaries of the feasible region do not remain in the population. Therefore, our CHDE requires a diversity mechanism (i.e., some infeasible solutions must remain in the population to avoid premature convergence) that does not increase its computational cost in a significant way.

With respect to the three state-of-the-art approaches, some facts require discussion: With respect to the Homomorphous Maps [16], our approach obtained a better "best" solution in nine problems (g01, g02, g03, g05, g06, g07, g09, g10 and g12) and a similar "best" results in other three (g04, h08 and g11). Also, CHDE provided a better "mean" result in five problems (g01, g05, g06, g08 and g12) and a better "worst" result for two problems (g05 and g12). It is clear that CHDE was superior in quality of results than the Homomorphous Maps and it was competitive based on statistical measures.

With respect to the Stochastic Ranking [9], CHDE was able to find a better "best" result in three problems (g02, g07 and g10) and a similar "best" result in the remaining ten problems (g01, g03, g04, g05, g06, g08, g09, g11, g12 and g13). Besides these, our approach got a similar "mean" and "worst" result for problem g12. CHDE found either similar or best quality results than the Stochastic Ranking, which is one of the most competitive approaches for evolutionary constrained optimization. However, SR is still more robust than CHDE. This is because SR has a good mechanism to maintain diversity in the population (keep both, feasible and infeasible solutions during all the process).

With respect to the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA) [18], our approach found better "best" results in three problems (g02, g07 and g10) and a similar "best" in eight functions (g01, g03, g04, g05, g06, g08, g09 and g11). Finally, CHDE could find a better "mean" result in problem g02. Our approach showed a competitive performance based on quality and showed some robustness compared to ASCHEA. However, the analysis was incomplete because the worst results found by ASCHEA were not available.

¿From the previous comparison, we can see that the CHDE produced competitive results based on quality with respect to three techniques representative of the state-of-the-art in constrained optimization. CHDE can deal with highly constrained problems, problems with low (g06 and g08) and high (g01, g02, g03, g07) dimensionality, with different types of combined constraints (linear, nonlinear, equality and inequality) and with very large (g02) or very small (g05, g13) or even disjoint (g12) feasible regions. However, our approach presented some robustness problems and more work is required in that direction.

It is worth emphasizing that CHDE does not require additional parameters. In contrast, the Homomorphous Maps require an additional parameter (called v) which has to be found empirically [16]. Stochastic ranking requires the definition of a parameter

		Best Result		Mean Result		Worst Result	
Problem	Optimal	CHDE	ASCHEA	CHDE	ASCHEA	CHDE	ASCHEA
g01	-15	-15.000	-15.0	-14.792134	-14.84	-12.743044	NA
g02	0.803619	0.803619	0.785	0.746236	0.59	0.302179	NA
g03	1	1.00	1.0	0.640326	0.99989	0.029601	NA
g04	-30665.539	-30665.539	30665.5	-30592.154435	30665.5	-29986.214382	NA
g05	5126.498	5126.496714	5126.5	5218.729114	5141.65	5502.410392	NA
g06	-6961.814	-6961.814	-6961.81	-6367.575424	-6961.81	-2236.950336	NA
g07	24.306	24.306	24.3323	104.599221	24.66	1120.541494	NA
g08	0.095825	0.095825	0.095825	0.091292	0.095825	0.027188	NA
g09	680.63	680.6300	680.630	692.472322	680.641	839.782911	NA
g10	7049.25	7049.248021	7061.13	8442.656946	7193.11	15580.370333	NA
g11	0.75	0.749	0.75	0.761823	0.75	0.870984	NA
g12	1	1	NA	1	NA	1	NA
g13	0.053950	0.053866	NA	0.747227	NA	2.259875	NA

Table 5. Comparison of our approach (CHDE) with respect to the Adaptive Segregational Constraint Handling Evolutionary Algorithm (ASCHEA). NA = Not Available.

called P_f, whose value has an important impact on the performance of the approach [9]. ASCHEA also requires the definition of several extra parameters, and in its latest version, it uses niching, which is a process that also has at least one additional parameter [18].

Measuring the computational cost, the number of fitness function evaluations (FFE) performed by our approach is lower than the other techniques with respect to which it was compared. Our approach performed 348,000 FFE. Stochastic ranking performed 350,000 FFE, the Homomorphous Maps performed 1,400,000 FFE, and ASCHEA performed 1,500,000 FFE.

7 Conclusions and Future Work

A novel approach based on the simplest version of the Differential Evolution algorithm, coupled with three simple criteria based on feasibility (CHDE) was proposed to solve constrained optimization problems. CHDE does not require a penalty function or any extra parameters (other than the original parameters of the DE algorithm) to bias the search towards the feasible region of a problem. Additionally, this improved approach has a low computational cost and it is easy to implement. Our algorithm was compared against three state-of-the-art techniques and it provided a competitive performance. Our future work consists on adding a diversity mechanism which does not increase its computational cost [8] in order to avoid premature convergence.

8 Acknowledgments

The first author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through a scholarship to pursue graduate studies at CINVESTAV-IPN's Electrical Engineering Department. The second author acknowledges support from (CONACyT) through projects number 32999-A and 34201-A.

References

1. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002) ISBN 0-3064-6762-3.
2. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Co., Reading, Massachusetts (1989)
3. Coello Coello, C.A.: Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. Computer Methods in Applied Mechanics and Engineering **191** (2002) 1245–1287
4. Michalewicz, Z., Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation **4** (1996) 1–32
5. Smith, A.E., Coit, D.W.: Constraint Handling Techniques—Penalty Functions. In Bäck, T., Fogel, D.B., Michalewicz, Z., eds.: Handbook of Evolutionary Computation. Oxford University Press and Institute of Physics Publishing (1997)
6. Price, K.V.: An Introduction to Differential Evolution. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. Mc Graw-Hill, UK (1999) 79–108
7. Mezura-Montes, E., Coello Coello, C.A.: A Simple Evolution Strategy to Solve Constrained Optimization problems. In: Proceedings of Genetic and Evolutionary Computation Conference(GECCO'2003), Heidelberg, Germany, Chicago, Illinois, USA, Springer-Verlag (2003) 641–642 Lecture Notes in Computer Science No. 2723.
8. Mezura-Montes, E., Coello Coello, C.A.: Adding a Diversity Mechanism to a Simple Evolution Strategy to Solve Constrained Optimization Problems. In: Proceedings of IEEE International Congress on Evolutionary Computation (CEC'2003), Canberra, Australia, IEEE Neural Networks Society (2003) (Accepted for Publication).
9. Runarsson, T.P., Yao, X.: Stochastic Ranking for Constrained Evolutionary Optimization. IEEE Transactions on Evolutionary Computation **4** (2000) 284–294
10. Schwefel, H.P.: Evolution and Optimal Seeking. John Wiley & Sons Inc., New York (1995)
11. Jiménez, F., Verdegay, J.L.: Evolutionary techniques for constrained optimization problems. In Zimmermann, H.J., ed.: 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany, Verlag Mainz (1999) ISBN 3-89653-808-X.
12. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering **186** (2000) 311–338
13. Coello Coello, C.A., Mezura-Montes, E.: Handling Constraints in Genetic Algorithms Using Dominance-Based Tournaments. In Parmee, I., ed.: Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002). Volume 5., University of Exeter, Devon, UK, Springer-Verlag (2002) 273–284
14. Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. Volume 1., Piscataway, New Jersey, IEEE Service Center (1994) 82–87
15. Storn, R.: System Design by Constraint Adaptation and Differential Evolution. IEEE Transactions on Evolutionary Computation **3** (1999) 22–34
16. Koziel, S., Michalewicz, Z.: Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. Evolutionary Computation **7** (1999) 19–44
17. Lampinen, J., Zelinka, I.: Mechanical Engineering Design Optimization by Differential Evolution. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. Mc Graw-Hill, UK (1999) 127–146
18. Hamida, S.B., Schoenauer, M.: ASCHEA: New Results Using Adaptive Segregational Constraint Handling. In: Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002). Volume 1., Piscataway, New Jersey, IEEE Service Center (2002) 884–889