Implementation of the Electromagnetism-like Algorithm with a Constraint-Handling Technique for Engineering Optimization Problems

Ana Maria A.C. Rocha and Edite M.G.P. Fernandes
Department of Production and Systems
University of Minho, Campus de Gualtar
4710-057 Braga, Portugal
{arocha;emgpf}@dps.uminho.pt

Abstract

This paper presents the implementation of a constraint-handling technique within the electromagnetism-like algorithm devised by Birbil and Fang in [1], for solving real-world engineering design problems. A derivative-free elite-based descent search scheme is also included in the final stage of each iteration to accelerate convergence and improve accuracy. Numerical experiments with a set of engineering problems are carried out and a comparison with other stochastic methods is reported.

Keywords: Global optimization, electromagnetism-like algorithm; constraint-handling; descent local search; engineering design problems.

1. Introduction

The problems that are addressed in this paper consider finding a global solution of a nonlinear optimization problem of the form:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leq 0, \ x \in \Omega, \end{array} \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ are nonlinear continuous functions and $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ is a closed set. We assume that the objective function f is nonconvex and may possess many local minima in the feasible region. This class of global optimization problems arises frequently in engineering applications. In the last decades, many algorithms have been proposed to solve problem (1). For large scale problems involving non-differentiable functions, the most used are stochastic-type algorithms. The following four main categories of methods to handle constraints in these algorithms are:

(1) Methods based on penalty functions. The constraints violation is combined with the objective function to define a penalty function. The optimization procedure is focussed on the penalty function. The main difficulty here is the updating of the penalty parameter. If this value is large, feasible solutions can be obtained although with low accuracy, since search around the boundary tends to be avoided. If the penalty parameter is small, infeasible but high quality solutions can be obtained. A dynamic updating of the penalty parameter is indeed the best strategy [9, 14].

(2) Methods based on multiobjective optimization concepts. Here, both the constraints violation and the objective function are objectives to be minimized by a multiobjective optimization method. However, solving multiobjective problems turns out to be more difficult and expensive that solving uniobjective problems [2, 5, 10].

(3) Methods based on biasing feasible over infeasible solutions. They seem to be nowadays the most used strategies for handling constraints. The constraints violation and the objective function are used separately and optimized by some sort of order, being the constraints violation the most important. This type of approach is possible in population-based methods but cannot be used with classical point-to-point search methods. See, for example, [3, 12, 13, 15].

(4) Methods based on preserving feasibility of solutions. Constraints are used to check if a particular point is feasible. Based on a population of feasible points, the method preserves feasibility. Infeasible points are discarded or repaired. In practice, the main difficulty is the generation of a population of feasible points, specially when equality constraints are present [6].

For a throughout review, we refer to [16] and the references therein reported. In this paper, we implement a constraint-handling technique for biasing feasible over infeasible

978-0-7695-3326-1/08 $25.00 © 2008 IEEE
DOI 10.1109/HIS.2008.145

solutions within the electromagnetism-like (EM) algorithm proposed by Birbil and Fang in [1], for solving real-world engineering design optimization problems. The technique to handle the constraints is employed since years in the field of multiobjective optimization and is efficient when used in a genetic algorithm context [3].

The remainder of this paper is organized as follows. The details that were implemented in the EM algorithm to handle the constraints are discussed in Section 2, as well as the new elite-based descent search. Section 3 contains the results of all the numerical experiments, including comparisons with other methods. Conclusions are made in Section 4.

2. The New Constrained EM Algorithm

In this section, we describe the new EM algorithm for solving problems of type (1). A simple to implement and efficient constraint-handling technique is included in the original EM algorithm devised by Birbil and Fang in [1] for bound constrained problems. A derivative-free elite-based descent search scheme is also included in the final stage of each iteration to accelerate convergence and improve accuracy. We first briefly list the main procedures of the EM algorithm.

2.1. EM algorithm for bound constraints

Here, we describe the EM algorithm for solving bound constrained problems [1]. This is a very simple population-based algorithm that simulates the electromagnetism theory of physics and uses an attraction-repulsion mechanism to move the population of points towards optimality. The algorithm starts with a population of randomly generated points from the feasible set Ω. Analogous to electromagnetism, each point in the space is considered as a charged particle. The charge of each point is related to the objective function value and determines the magnitude of attraction of the point over the others in the population. The better the objective function value, the higher the magnitude of attraction. The charges are used to find the total force exerted on each point, by the other points in the population, as well as a direction to move the points for the subsequent iteration. A random local search is also included to refine the search around one point of the population. We refer to [1] for details.

2.2. The constraint-handling technique

Most stochastic methods for global optimization are developed primarily for unconstrained problems. Then they are extended to constrained optimization problems by modifying the original procedures or by using penalty functions. Here, we convert the EM method for bound constrained optimization problems [1] into an algorithm for constrained optimization using a method based on biasing feasible over infeasible solutions [3].

The degree of constraints violation that is herein adopted considers $G(x) = \sum_{j=1}^{p} [g_j(x)]_+$ where

$$[g_j(x)]_+ = \max\{0, g_j(x)\}, \quad 1 \leq j \leq p. \tag{2}$$

We remark that at a feasible point $G(x) = 0$. The selection of the most promising points in the population is based on a fitness concept that relies on the following (fitness) function, proposed in [3],

$$\phi(x) = \begin{cases} f(x) & \text{if } G(x) = 0, \\ f_{\max} + G(x) & \text{otherwise,} \end{cases} \tag{3}$$

where $f_{\max}$ is the objective function value of the worst feasible point in the population. If no feasible solution exists, in a particular iteration, $f_{\max}$ is set to 0.

We remark that the adopted constraint-handling technique does not require any penalty parameter since points in the population are not compared in terms of both objective function value and constraints violation. Thus, when comparing pairwise points, one case of the three below is operated:

(1) among two feasible points, the one that has better objective function value is preferred;

(2) any feasible point is preferred to any infeasible point;

(3) among two infeasible points, the one that has smaller constraints violation is preferred.

The remaining part of this section presents a detailed description of each main procedure in the new EM algorithm.

2.3. Initialization of the population

The algorithm starts with a population of p_{size} points randomly generated from the set Ω. The ith point of the population is denoted by x^i. Each coordinate of a point is represented by x_k^i ($k = 1, \ldots, n$) and is computed using $x_k^i = l_k + \lambda(u_k - l_k)$ where $\lambda \sim U(0, 1)$. Constraints violation or/and objective function value for all points are evaluated and a pairwise comparison is carried out in order to identify the best point, x^{best}. In practice, we first check the feasibility of all points in the population. If the point is feasible, the objective function value is evaluated. On the other hand, if the point is infeasible, its objective function value is not required.

2.4. Charge and force vector calculation

The total force vector F^i exerted on each point x^i by the other $p_{size} - 1$ points is calculated by adding the individual component forces, F_j^i, between any pair of points x^i and x^j of the population. According to the electromagnetism theory, each individual force is inversely proportional to the square of the distance between the two points and directly proportional to the product of their charges.

The charge q^i of point x^i determines the power of attraction or repulsion for that point and is computed according to the fitness function value by

$$q^i = \exp\left(\frac{-n(\phi(x^i) - \phi(x^{best}))}{\phi(x^{worst}) - \phi(x^{best})}\right), \qquad (4)$$

for $i = 1, \ldots, p_{size}$, where x^{worst} represents the point in the population with the worst fitness function value. Note that the points with better fitness function values possess higher charges. Feasible points will always have higher charges than the infeasible ones. Charges in (4) are based on the distance of the fitness value at x^i to the fitness value of the best point in the population, scaled by the range of values. Tests with different charge formulas show the efficiency of this proposal [11].

For the positive charges (4), the direction of a force F_j^i depends on a pairwise comparison with the points x^i and x^j. Hence, if $\phi(x^j) < \phi(x^i)$, the point x^j attracts x^i and consequently the direction of the force is $\overrightarrow{x^i x^j}$. Otherwise, x^j repels x^i and the direction of the force is $\overrightarrow{x^j x^i}$. Thus,

$$F_j^i = \begin{cases} (x^j - x^i)\frac{q^i q^j}{\|x^j - x^i\|^3} & \text{if } \phi(x^j) < \phi(x^i) \\ (x^i - x^j)\frac{q^i q^j}{\|x^j - x^i\|^3} & \text{otherwise,} \end{cases} \qquad (5)$$

for $j \neq i$. Then the total force vector F^i exerted on each point x^i by the other $p_{size} - 1$ points is calculated by adding the individual component forces as follows: $F^i = \sum_{j\neq i}^{p_{size}} F_j^i$, $i = 1, \ldots, p_{size}$.

2.5. Move points

The total force vector, F^i, is used to move the point x^i in the direction of the force by a random step length $\lambda \sim U(0, 1)$ as follows

$$x_k^i = \begin{cases} x_k^i + \lambda \frac{F_k^i}{\|F^i\|}(u_k - x_k^i) & \text{if } F_k^i > 0 \\ x_k^i + \lambda \frac{F_k^i}{\|F^i\|}(x_k^i - l_k) & \text{otherwise,} \end{cases} \qquad (6)$$

for each k ($k = 1, 2, \ldots, n$) and for $i = 1, \ldots, p_{size}$ and $i \neq best$. Note that the best point, x^{best}, is not moved. Using the normalized total force vector and scaling it with the allowed range of movement towards the lower bound l_k, or the upper bound u_k, x^i remains in the set Ω. Based on the fitness function values, the best point in the population is then identified.

2.6. Elite-based descent search

Like other stochastic algorithms, this new EM algorithm needs a local search procedure to accelerate the convergence speed in the final stage of each iteration. Furthermore, local search models aim to improve the accuracy of the solution. Local search can exploit regions around all points, a selected set of points, or only one point in the population. Here, we use an elitist approach by refining a predefined region of the selected best point in the population. Thus, this elite-based search scheme is used to improve accuracy of the best solution found so far.

The objective is to generate a sequence of approximations of the optimizer by computing approximate descent directions, starting from the best point, within a predefined number of iterations. The approach to compute the approximate descent directions is similar to the one in [5]. However, the herein proposed local search uses the fitness function (3) for the descent purposes and for pairwise comparisons.

A brief description of this elite-based approach is now presented. First, two exploring points x_i^{rand}, $i = 1, 2$, in a neighborhood of the best point x^{best}, of ray ε_r (positive), are randomly generated. The approximate descent direction, d, for the fitness function ϕ, at the best point x^{best}, is then generated by

$$d = -\frac{1}{\sum_{j=1}^{2} |\Delta\phi_j|} \sum_{i=1}^{2} \Delta\phi_i \frac{x^{best} - x_i^{rand}}{\|x^{best} - x_i^{rand}\|}, \qquad (7)$$

where $\Delta\phi_j = \phi(x^{best}) - \phi(x_j^{rand})$. We refer to [5] for details and theoretical properties.

The best point is moved along this descent direction, d, with a step size of α, where $0 < \alpha \leq 1$. We remark that the descent direction (7) is normalized and scaled with the allowed range of movement towards the lower bound l_k, or the upper bound u_k, of the set Ω, as proposed in (6), to maintain bound constraints feasibility. Let d_{norm} be the normalized and scaled direction.

This local search procedure also implements a backtracking strategy to progress towards optimality, as follows. After a new point $y = x_{best} + \alpha d_{norm}$ has been computed, the fitness function is used to compare y and x_{best}. If y is preferred to x_{best}, then y replaces x_{best} and the search is repeated, evaluating a new descent direction and re-initializing the step size α to 1. On the other hand, if x^{best} is the preferred point, then y is discarded, the step size is halved (*i.e.*, $\alpha \leftarrow \alpha/2$) and a new point is evaluated along the same descent direction. This process is repeated at least for nls_{max} iterations.

3. Numerical Experiments

Problems of practical interest are important for assessing the effectiveness of a given approach. Thus, to evaluate the performance of the herein proposed EM algorithm for constrained problems, a set of 7 benchmark engineering problems is used. Comparisons with other published results are also included. The algorithm is coded in the C programming language. The code also includes an interface to be able to read problems coded in AMPL [4]. The set of coded problems may be obtained from the first author upon request.

3.1. Engineering problems

We now summarize the characteristics of the chosen engineering problems. They all have bound and inequality constraints. These problems are fully described in the below cited references.

(1) Design of a welded `beam` [5, 7, 8, 10]. In this problem, the cost of a welded beam is minimized, subject to constraints on the shear stress, bending stress in the beam, buckling load on the bar, end deflection of the beam, and side constraints. There are 4 design variables (h, l, t and b), and 7 inequality constraints.

(2) Design of a `heat` exchanger [3, 7, 8]. This problem involves minimizing the sum of the heat transfer areas of three exchangers. There are 8 design variables (A_1, A_2, A_3, T_1, T_2, t_{12}, t_{22} and t_{32}), and 6 inequality constraints.

(3) Design of a `speed` reducer [8, 10]. The objective in this problem is to minimize the total weight of a speed reducer, subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. There are 7 design variables (b, m, z, l_1, l_2, d_1 and d_2), and 11 inequality constraints.

(4) Design of a tension/compression `spring` [5, 6, 8, 10]. This problem minimizes the weight of a tension/compression spring, subject to constraints on the minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The problem has 3 design variables (d, D and N), and 4 inequality constraints.

(5) Design of three-bar `truss` [10]. This problem minimizes the volume of a 3-bar truss structure, subject to stress constraints. The problem has 2 design variables (A_1 and A_2) representing cross-sectional areas of two bars (since $A_1 = A_3$), and 3 inequality constraints.

(6) Design of a `tubular` column [8]. The objective in this problem is to minimize the total cost of the material and construction of a tubular column. The problem has 2 design variables (d and t), and 2 inequality constraints.

(7) Design of a pressure `vessel` [5, 6, 7]. This problem consists of minimizing the total cost of the material, forming and welding of a cylindrical pressure vessel. The problem has 4 design variables (T_s, T_h, R and L) and 4 inequality constraints.

3.2. Comparative results

As required by any stochastic algorithm, we record values of the best function value, f_{best}, the average of the best function values, f_{avg}, and the standard deviation, SD, obtained after 50 independent runs, each starting from a random population with different seeds. Here, we use a limit of $Nit_{max} = 1000$ iterations to terminate the algorithm. The subsequent tables also report the average numbers of objective function and constraint evaluations, nf_{eval} and nc_{eval}, respectively. We consider a population size of $10n$ points, except otherwise stated. The selected parameters in the new EM algorithm are as follows: $nls_{max} = 10$ and $\varepsilon_r = 0.001$.

Table 1. Results for the beam problem.

	Best solution		
	new EM	in [5]	in [10]
h	0.243738	0.244353	0.244438
l	6.247215	6.215792	6.237967
t	8.290347	8.293904	8.288576
b	0.244477	0.244353	0.244566
f_{best}	2.384292	2.381065	2.385435
f_{avg}	2.434450	2.404166	3.255137
SD	0.025991	-	0.959078
nf_{eval}	31751	56243	33095
nc_{eval}	51087	23989	-

First, we deal with the welded beam design problem. A comparison of results is presented in Table 1. The results regarding the other methods are taken from the original papers [5, 10]. The method in [5], is a simulated annealing method of a multi-start type, which uses a filter technique. The method proposed in [10] is a population-based method that is based on the concept of society and civilization. The results presented in [10] use a population of $10n$ points and correspond to 50 runs. The results reported in [5] for this problem consider 50 runs. Our results are competitive. For a similar number of function evaluations we obtain a best solution of 2.384292, with a SD of 0.025991, slightly

Table 2. Results for the speed problem.

	Best solution		
	new EM	in [10]	in [16]
b	3.500885	3.500007	3.500023
m	0.700137	0.700000	0.700000
z	17	17	17.00001
l_1	7.505659	7.327602	7.300428
l_2	7.790018	7.715322	7.715377
d_1	3.350843	3.350267	3.350231
d_2	5.286704	5.286655	5.286664
f_{best}	2998.9668	2994.7442	2994.4991
f_{avg}	3015.0381	3001.7583	2994.6134
SD	8.56217	4.00914	7.0e-02
nf_{eval}	3711	54456	40000
nc_{eval}	81149	-	-
p_{size}	70	70	100

Table 3. Results for the spring problem.

	Best solution		
	new EM	in [5]	in [10]
d	0.051559	0.051743	0.052160
D	0.353596	0.358005	0.368159
N	11.47500	11.21391	10.64844
f_{best}	0.012666	0.012665	0.012669
f_{avg}	0.012697	0.012665	0.012923
SD	2.9e-05	2.2e-08	5.9e-04
nf_{eval}	6596	49531	25167
nc_{eval}	41077	18802	-

below the best value reached in [10]. However, the best value 2.381065 attained by [5] is better than ours.

When solving the heat exchanger design problem, [3] reaches the best value 7063.377 after 80080 function evaluations, with a population size of 80 and 1000 maximum generations. In the paper, the author reached the value 7060.221 (a value about 0.15% worse than the true optimal value 7049.330923), within 4000 maximum generations and 320080 function evaluations. The harmony search meta-heuristic algorithm of [7] found 20 different solution vectors with the best value of 7057.274414 after 150000 searches. Our EM algorithm solved the heat problem (with a population of 80 points) and after 1000 iterations the best solution found is 7432.696. The average number of objective function evaluations is 2644 and the average number of constraint evaluations is 91415. When we use $Nit_{max} = 4000$, the best solution found is 7156.23 (1.5% above the true optimum), with $nf_{eval} = 9778$ and $nc_{eval} = 364465$. We also allowed our algorithm to reach 80000 function evaluations and after 50 runs, the best found solution was 7062.087 (0.18% above the true optimum).

We now report the results obtained when solving the speed reducer design problem. Table 2 compares our EM algorithm with other stochastic algorithms. The results reported in [10] correspond to 50 runs and the population contains 70 points. In [16], a hybrid evolutionary algorithm with an adaptive constraint-handling technique is proposed. The results therein registered are obtained after 30 runs. With a population of 70 points our best result is 0.15% worst than the best in the table (2994.4991 in [16]). When the new EM is run with a population of 140 points we obtain: $f_{best} = 3002.5357$ (0.3% worst than 2994.4991) with $nf_{eval} = 13679$ and $nc_{eval} = 151218$.

Table 4. Results for the truss problem.

	Best solution		
	new EM	in [10]	in [16]
A_1	0.789428	0.788621	0.788680
A_2	0.406133	0.408401	0.408234
f_{best}	263.8973	263.8958	263.8958
f_{avg}	263.9082	263.9033	263.8959
SD	5.3e-03	1.3e-2	4.9e-05
nf_{eval}	8887	17610	15000
nc_{eval}	31025	-	-
p_{size}	20	20	100

Table 5. Results for the tubular problem.

	Best solution		
	new EM	new EM	in [8]
d	5.451238	5.450971	5.4507
t	0.291965	0.291999	0.2920
f_{best}	26.53171	26.53222	25.5316
f_{avg}	26.53879	26.54295	-
SD	4.6e-03	9.4e-3	-
nf_{eval}	38418	20936	-
nc_{eval}	51054	31034	55
nit_{max}	1000	1000	53
p_{size}	40	20	-

Table 3 presents the results obtained by the new EM algorithm when solving the spring problem, after 50 runs with a population of 30 points. A comparison with the results in [5] and [10] is made. The results in [5] were obtained after 30 runs. Although our best value is not better than the solution found in [5], it is better than the one in [10]. When a population of 60 points is used, our best results are as follows: $f_{best} = 0.012666$, $SD = 1.9e-05$, $nf_{eval} = 17600$ and $nc_{eval} = 71107$.

In Table 4 we register our best results with truss problem obtained after 50 runs with a population of 20 points (conditions similar to [10]). In [16], the results correspond

Table 6. Results for the vessel problem.

	Best solution		
	new EM	in [5]	in [7]
T_s	0.813263	0.768326	1.125
T_h	0.402966	0.379784	0.625
R	42.131037	39.80962	58.2789
L	176.2978	207.2256	43.7549
f_{best}	5953.519	5868.765	7198.433
f_{avg}	6264.979	6164.586	-
SD	195.675	257.474	-
nf_{eval}	31221	108883	-
nc_{eval}	51095	49253	-
p_{size}	40	-	20

to 30 runs with a population of 100 points. Our EM algorithm is competitive. When the population is increased to 40 points, the following results are obtained: $f_{best} = 263.8989$, $SD = 4.6e-03$, $nf_{eval} = 16782$ and $nc_{eval} = 51045$.

Table 5 lists the best results obtained by the new EM for 40 and 20 points in the population (50 runs) when solving the tubular problem. Results obtained with a fuzzy proportional-derivative controller method proposed in [8] are listed for comparison.

Our algorithm achieves a design for the vessel problem with a best solution of 5953.519, much better than the optimal value found in [7], and 1.4% worst than the solution in [5] - found after 30 runs. See Table 6. If we use a population of 80 points, the best found solution is 5945.445 with a maximum of 61772 function evaluations and 91133 constraint evaluations.

4. Conclusions

This paper presents the implementation of an efficient constraint-handling technique, previously tested in [3] in a genetic algorithm context, within the EM algorithm, for solving constrained global optimization problems. Charge and force vector calculations rely on a fitness function that assigns better function values to feasible than to infeasible points. The algorithm also incorporates an elite-based search scheme that relies on a descent direction to provide high accuracy solutions. Tests with a set of engineering design problems and comparisons with recently proposed methods are included. The experimental results indicate that the selected problems can be efficiently solved by this new EM algorithm. Our study will continue addressing other interesting and more complex engineering problems, such as larger truss sizing and configuration optimization problems and short-term electric hydrothermal scheduling problems.

References

[1] S. I. Birbil and S.-C. Fang. An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25:263–282, 2003.

[2] C. A. Coello Coello. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering and Environmental Systems*, 17:319–346, 2000.

[3] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338, 2000.

[4] R. Fourer, D. M. Gay, and B. W. Kernighan. A modeling language for mathematical programming. *Management Science*, 36(5):519–554, 1990.

[5] A.-R. Hedar and M. Fukushima. Derivative-free filter simulated annealing method for constrained continuous global optimization. *Journal of Global Optimization*, 35:521–549, 2006.

[6] X. Hu, R. C. Eberhart, and Y. Shi. *Engineering optimization with particle swarm*. Proceedings of 2003 IEEE Swarm Intelligence Symposium, 53–57, 2003.

[7] K. S. Lee and Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 194:3902–3933, 2005.

[8] T.-C. Liu. *Developing a fuzzy proportional-derivative controller optimization engine for engineering optimization problems*. PhD thesis, August 2006.

[9] Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis. Memetic particle swarm optimization. *Annals of Operations Research*, 156:99–127, 2007.

[10] T. Ray and K. M. Liew. Society and Civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, 7(4):386–396, 2003.

[11] A. M. A. C. Rocha and E. M. G. P. Fernandes. On charge effects to the electromagnetism-like algorithm. EurOPT 2008, L.Sakalauskas et al. (eds.) ISBN 978-9955-28-283-9, 198–203, 2008.

[12] A. M. A. C. Rocha and E. M. G. P. Fernandes. Feasibility and dominance rules in the electromagnetism-like algorithm for constrained global optimization problems. *Lecture Notes in Computer Science*, Computational Science and Its Applications (O. Gervasi et al. (eds.)), 5073, 768-783, 2008.

[13] T. P. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, 2000.

[14] M.-J. Tahk, H.-W. Woo, and M.-S. Park. A hybrid optimization method of evolutionary and gradient search. *Engineering Optimization*, 39(1):87–104, 2007.

[15] T. Takahama and S. Sakai. Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations. *IEEE Transactions on Evolutionary Computation*, 9(5):437–451, 2005.

[16] Y. Wang, Z. Cai, Y. Zhou and Z. Fan. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Structural Multidisciplinary Optimization*, DOI 10.1007/s00158–008–0238–3.