A New Approach for Solving Nonlinear Equations Systems

Crina Grosan and Ajith Abraham, *Senior Member, IEEE*

***Abstract*—This paper proposes a new perspective for solving systems of complex nonlinear equations by simply viewing them as a multiobjective optimization problem. Every equation in the system represents an objective function whose goal is to minimize the difference between the right and left terms of the corresponding equation. An evolutionary computation technique is applied to solve the problem obtained by transforming the system into a multiobjective optimization problem. The results obtained are compared with a very new technique that is considered as efficient and is also compared with some of the standard techniques that are used for solving nonlinear equations systems. Several well-known and difficult applications (such as interval arithmetic benchmark, kinematic application, neuropsychology application, combustion application, and chemical equilibrium application) are considered for testing the performance of the new approach. Empirical results reveal that the proposed approach is able to deal with high-dimensional equations systems.**

***Index Terms*—Computational intelligence, evolutionary multiobjective optimization, metaheuristics, nonlinear equation systems.**

I. Introduction

Systems of nonlinear equations arise in many domains of practical importance such as engineering, mechanics, medicine, chemistry, and robotics. Solving such a system involves finding all the solutions (there are situations when more than one solution exists) of the polynomial equations contained in the mentioned system. The problem is nondeterministic polynomial-time hard, and it is having very high computational complexity due to several numerical issues [27]. There are several approaches for solving these types of problems. Van Hentenryck *et al.* [27] divided these approaches into two main categories:

1) interval methods that are generally robust but tend to be slow;
2) continuation methods that are effective for problems for which the total degree is not too high [27].

The limitations of Newton's method are pointed out in the aforementioned works. Bader [5] mentioned that standard direct methods, such as Newton's method, are impractical for large-scale problems because of their high linear algebra costs and large memory requirements. Bader proposed a tensor method using Krylov subspace methods for solving large-scale systems of linear equations. There is a condition to be fulfilled—the equations must be continuously differentiable at least once. Bader's paper also provides a good review of similar research for solving systems of equations.

Krylov subspace methods based on moment matching are also used by Salimbahrami and Lohmann [44]. Effati and Nazemi [18] proposed a very efficient approach for solving nonlinear systems of equations. Although there are several existing approaches for solving systems of nonlinear equations, there are still limitations of the existing techniques, and, still, more research is to be done.

There is a class of methods for the numerical solutions of the above system, which arise from iterative procedures used for systems of linear equations [39], [43]. These methods use reduction to simpler 1-D nonlinear equations for the components $f_1, f_2, \ldots, f_n$ [26]. In a strategy based on trust regions [30], at each iteration, a convex quadratic function is minimized to determine the next feasible point to step to. The convex quadratic function is the squared norm of the original system plus a linear function multiplied by the Jacobian matrix. There is also the approach of homotopy methods, which are sometimes referred to as continuation methods [28], [30], [38]. This approach begins with a "starting" system of equations (not the true system) whose solution is known. This starting system is gradually transformed to the original system. At each stage, the current system is solved to find a starting solution for the next stage system. The idea is that as the system changes, the solutions trace out a path from a solution of the starting system to a solution of the original system. At each stage, the current system is normally solved by a Newton-type method [28]. The dimension reducing method, the modified reducing dimension method, and the perturbed dimension reducing method [21]–[25] are also methods for numerical solutions of systems of nonlinear equations, which incorporate Newton and nonlinear successive overrelaxation algorithms [39] and use reduction to simpler 1-D nonlinear equations (but they quadratically converge).

In the approach proposed in [36], the system of equations is transformed into a constraint optimization problem. At each step, some equations that are satisfied at the current point are treated as constraints and the other ones as objective functions. The set $\{1, 2, \ldots, n\}$ is divided into two parts, i.e., S_1 and S_2, where S_2 denotes the complement $\{1, 2, \ldots, n\} \setminus S_1$. Then, the problem is given by

$$\text{minimize } \sum_{i \in S_1} f_i^2(x)$$
$$\text{subject to } f_j(x) = 0, \quad j \in S_2.$$

Manuscript received September 9, 2006; revised March 22, 2007. This paper was recommended by Associate Editor J. Yang.

C. Grosan is with the Department of Computer Science, Babes-Bolyai University, 400084 Cluj Napoca, Romania (e-mail: cgrosan@cs.ubbcluj.ro).

A. Abraham is with Center of Excellence for Q2S, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: ajith.abraham@ieee.org).

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/TSMCA.2008.918599

The equations system is reduced to the same form in the approach used in [37].

The optimization problem obtained in [28] by transforming the systems of equations is similar to the one proposed in [36] and considers the equation given by the sum of squared components $f_1, f_2, \ldots, f_n$.

In this paper, we propose a novel approach that transforms a system of nonlinear equations into a multiobjective optimization problem. The new obtained problem is solved using the standard Pareto dominance relationship between solutions and an iterative strategy that evolves some random solutions in the search for optimal solutions. The technique uses principles from the evolutionary computation field and is able to approximate the solutions even for large-scale systems of equations. Moreover, no additional constraints involving extra problem knowledge (such as the condition that equations must be differentiable) are required.

The successful application of the multiobjective optimization approaches is well known, and there is a huge amount of work in this field reporting applications in different domains in science and engineering [1], [11].

Abido [2] developed Pareto-based multiobjective evolutionary algorithms for solving a real-world power system multiobjective nonlinear optimization problem. Specifically, the nondominated sorting genetic algorithm (NSGA), the niched Pareto genetic algorithm, and the strength Pareto evolutionary algorithm were developed and successfully applied to an environmental/economic electric power dispatch problem.

Benedetti *et al.* [6] illustrated that when dealing with the multiobjective optimization of the tire suspension system of a racing car, a large number of design variables and a large number of objectives have to be taken into account. Two different models have been used, which are both validated on data coming from an instrumented car—a differential equation-based model and a neural network model. Up to 23 objective functions have been defined, and at least 14 of which were in strict conflict of each other. Benedetti *et al.* provided a fuzzy definition of optima, being a generalization of Pareto optimality, and the result of such an approach is that subsets of Pareto optimal solutions can be properly selected as a consequence of input from the designer. The obtained optimal solutions were compared with the reference vehicle and with the optima previously obtained with the design of experiment techniques and different multiobjective optimization strategies.

Tan *et al.* [49] developed a cooperative coevolutionary algorithm (CCEA) for multiobjective optimization, which applies the divide-and-conquer approach to decompose decision vectors into smaller components and evolves multiple solutions in the form of cooperative subpopulation. Incorporated with various features like archiving, dynamic sharing, and an extending operator, the CCEA is capable of maintaining archive diversity in the evolution and uniformly distributing the solutions along the Pareto front. Exploiting the inherent parallelism of cooperative coevolution, the CCEA can be formulated into a distributed CCEA that is suitable for concurrent processing that allows the intercommunication of subpopulation residing in networked computers and, hence, expedites the computational speed by sharing the workload among multiple computers.

Deb *et al.* [12] used the NSGA II for the optimization of the epoxy polymerization process. The problem is a well-known chemical engineering problem and involves the optimization of three conflicting objectives and 20 variables. A modified differential evolution is used by Babu *et al.* [4] and Angira and Babu [3] for solving practical multiobjective optimization problems from chemistry.

Medaglia *et al.* [31] proposed an evolutionary method for project selection problems with partially funded projects, multiple (stochastic) objectives, project interdependence (in the objectives), and a linear structure for resource constraints. The method is based on the posterior articulation of preferences and is able to approximate the efficient frontier composed of stochastically nondominated solutions.

Chen *et al.* [9] developed an efficient macroevolutionary multiobjective genetic algorithm (MMGA) for optimizing the rule curves of a multipurpose reservoir system in Taiwan. Macroevolution is a new kind of high-level species evolution that can avoid premature convergence that may arise during the selection process of conventional genetic algorithms. The MMGA enriches the capabilities of genetic algorithms to handle multiobjective problems by diversifying the solution set.

Monitoring complex environmental systems is extremely challenging because it requires environmental professionals to capture impacted systems' governing processes, elucidate human and ecologic risks, limit monitoring costs, and satisfy the interests of multiple stakeholders (e.g., site owners, regulators, and public advocates). Reed *et al.* [42] illustrated how evolutionary multiobjective optimization has tremendous potential to help resolve these issues by providing environmental stakeholders with a direct understanding of their monitoring tradeoffs. Reed *et al.* used dominance archiving and automatic parameterization techniques to significantly improve the ease of use and efficiency of multiobjective optimization algorithms.

Heavy industry maintenance facilities at aircraft service centers or railroad yards must contend with scheduling preventive maintenance tasks to ensure that critical equipment remains available. All preventive maintenance tasks should be completed as quickly as possible to make the equipment available. This means that the completion time should be also minimized. A cost-effective schedule should strike some balance between a minimum schedule and a minimum size workforce. Quan *et al.* [41] used evolutionary algorithms to solve this multiobjective problem. Rather than conducting a conventional dominance-based Pareto search, Quan *et al.* introduced a form of utility theory to find the Pareto optimal solutions. The advantage of this method is that the user can target specific subsets of the Pareto front by merely ranking a small set of initial solutions.

The performance of the proposed approach is evaluated for several well-known benchmark problems from kinematics, chemistry, combustion, and medicine. Numerical results reveal the efficiency of the proposed approach and its flexibility to solve large-scale systems of equations.

II. Basic Concepts

A nonlinear system of equations is defined as

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

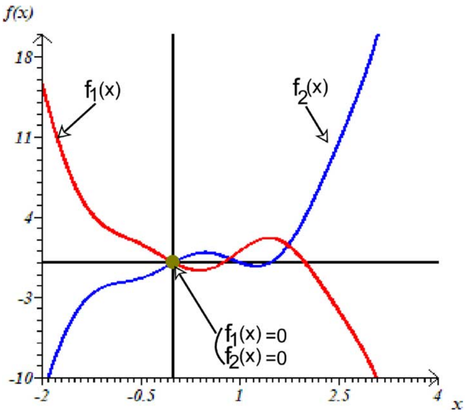

Fig. 1. Example of a solution for two nonlinear equations systems represented by f_1 and f_2.

where $x = (x_1, x_2, \ldots, x_n)$ refers to n equations and n variables, and $f_1, \ldots, f_n$ are nonlinear functions in the space of all real valued continuous functions on $\Omega = \prod_{i=1}^{n}[a_i, b_i] \subset \Re^n$. Some of the equations can be linear, but not all of them. Finding a solution for a nonlinear system of equations $f(x)$ involves finding a solution such that every equation in the nonlinear system is 0, i.e.,

$$(P)\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \quad\vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0. \end{cases} \tag{1}$$

In Fig. 1, the solution for a system having two nonlinear equations is depicted.

There are also situations when a system of equations is having multiple solutions. For instance, the system

$$\begin{cases} f_1(x_1, x_2, x_3, x_4) = x_1^2 + 2x_2^2 + \cos(x_3) - x_4^2 = 0 \\ f_2(x_1, x_2, x_3, x_4) = 3x_1^2 + x_2^2 + \sin^2(x_3) - x_4^2 = 0 \\ f_3(x_1, x_2, x_3, x_4) = -2x_1^2 - x_2^2 - \cos(x_3) + x_4^2 = 0 \\ f_4(x_1, x_2, x_3, x_4) = -x_1^2 - x_2^2 - \cos^2(x_3) + x_4^2 = 0 \end{cases}$$

has two solutions: (1, −1, 0, 2) and (−1, 1, 0, −2). The assumption is that a zero, or root, of the system exists. The solutions we are interested in are those points (if any) that are common to the zero contours of f_i, $i = 1, \ldots, n$. There are several ways to solve nonlinear equations systems [7], [13]–[17], [39]. Probably the most popular techniques are the Newton-type techniques. Some other techniques are as follows:

- trust-region method [10];
- Broyden method [8];
- secant method [16];
- Halley method [39].

The *quasi-Newton methods* are similar to the conjugate gradient methods. The goal is to accumulate information from successive line minimizations so that N such line minimizations lead to the exact minimum of a quadratic form in N dimensions [40].

1) Newton's Method: We can approximate f by the first-order Taylor expansion in a neighborhood of a point $x^k \in \Re^n$. The Jacobian matrix $J(x^k) \subset \Re^{n \times n}$ for $f(x)$ evaluated at x^k is given by

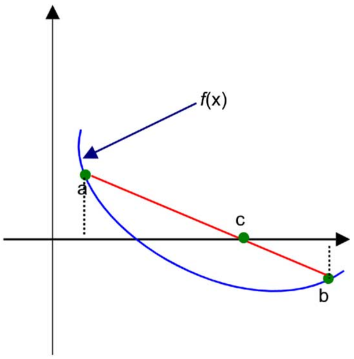

Fig. 2. Example of the secant method.

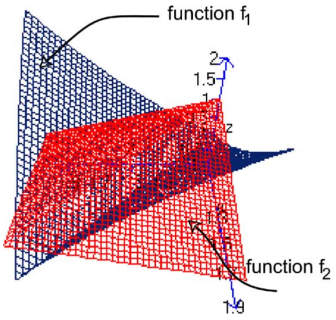

Fig. 3. Illustrative example.

TABLE I
PARAMETERS USED BY THE EVOLUTIONARY APPROACH

Parameter	Value
Population size	200
External set size	50
Number of generations	150
Sigma for mutation	0.23
Tournament size	5

$$J = \begin{bmatrix} \frac{\delta f_1}{\delta x_1} \cdots \frac{\delta f_1}{\delta xn} \\ \vdots \qquad\quad \vdots \\ \frac{\delta f_n}{\delta x_1} \cdots \frac{\delta f_n}{\delta x_n} \end{bmatrix}.$$

Then, we have

$$f(x^k + t) = f(x^k) + J(x^k)t + O\left(\|p\|^2\right).$$

By setting the right side of the equation to zero and neglecting terms of order higher than the first $[O(\|p\|^2)]$, we obtain

$$J(x^k)t = -f(x^k).$$

Then, the Newton algorithm is described as follows:

Algorithm 1 Newton algorithm.
Set $k = 0$.
Guess an approximate solution x^0.

TABLE II
PARETO FRONT OBTAINED BY THE EVOLUTIONARY APPROACH

Values			
x	y	f_1	f_2
0.996688	-1.000076	0.006765	0.0032346
0.991807	-0.994134	0.004620	0.0140101
0.991597	-0.991074	0.001036	0.0172525
-1.018355	1.0182264	0.000262	0.0369164
-0.983186	-1.016728	0.067080	0.0003659
-1.023594	-0.976637	0.093923	0.0003195
-1.001930	-1.004958	0.006077	0.0068986
-1.005263	-0.994158	0.022202	0.0006085
-1.027410	-1.027403	0.000014	0.0555657

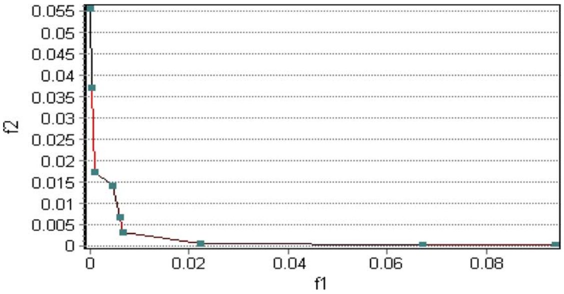

Fig. 4. Pareto front obtained by the evolutionary approach.

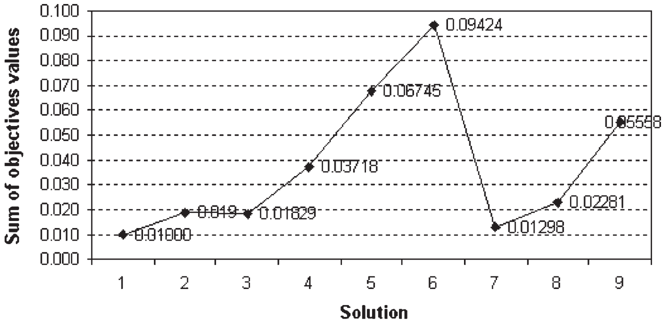

Fig. 5. Sum of the absolute values of the nondominated solutions obtained.

```
Repeat
    Compute $J(x^k)$ and $f(x^k)$.
    Solve the linear system $J(x^k)t = -f(x^k)$.
    Set $x^{k+1} = x^k + t$.
    Set $t = t + 1$.
Until converge to the solution
```

The index k is an iteration index, and x^k is the vector x after k iterations. The idea of the method is to start with a value that is reasonably close to the true zero, then replace the function by its tangent, and compute the zero of this tangent. This zero of the tangent will typically be a better approximation to the function's zero, and the method can be iterated.

Remarks:

1) This algorithm is also known as the Newton–Raphson method. There are also several other Newton methods.
2) The algorithm converges fast to the solution.
3) It is very important to have a good starting value (the success of the algorithm depends on this).

TABLE III
VALUES OF THE PARAMETERS USED IN THE EXPERIMENTS BY THE EVOLUTIONARY APPROACH

Parameter	Value	
	Example 1	Example 2
Population size	250	300
External set size	100	100
Number of generations	150	200
Sigma (for mutation)	0.1	0.1
Tournament size	4	5

TABLE IV
RESULTS FOR THE FIRST EXAMPLE

Method	Solution	Functions values
Newton's method	(0.15, 0.49)	(-0.00168, 0.01497)
Secant method	(0.15, 0.49)	(-0.00168, 0.01497)
Broyden's method	(0.15, 0.49)	(-0.00168, 0.01497)
Effati's method	(0.1575, 0.4970)	(0.005455, 0.00739)
Evolutionary approach	(0.15772, 0.49458)	(0.001264, 0.000969)

TABLE V
RESULTS FOR THE SECOND EXAMPLE

Method	Solution	Functions values
Effati	(0.0096, 0.9976)	(0.019223, 0.016776)
Evolutionary approach	(-0.00138, 1.0027)	(-0.00276, -6,37E-5)

TABLE VI
BENCHMARKS USED IN THE EXPERIMENTS

Benchmark		Number of variables	Range
1.	Interval i1	10	[-2, 2]
2.	Neurophysiology application	6	[-10, 10]
3.	Chemical equilibrium	5	[-10, 10]
4.	Kinematics kin2	8	[-10, 10]
5.	Combustion application	10	[-10, 10]
6.	Economics e2	20	[-10, 10]

TABLE VII
PARAMETERS USED BY THE EVOLUTIONARY APPROACH FOR THE INTERVAL ARITHMETIC I1 BENCHMARK

Parameter	Value
Population size	500
External set size	200
Number of generations	300
Sigma (for mutation)	0.1
Tournament size	5

4) The Jacobian matrix is needed; however, in many problems, analytic derivatives are unavailable.
5) If function evaluation is expensive, then the cost of finite-difference determination of the Jacobian can be prohibitive.

2) Broyden's Method: Let us denote [40] the approximate Jacobian by B and let

$$\delta x = -J^{-1} f.$$

TABLE VIII
EXAMPLES OF NONDOMINATED SOLUTIONS OBTAINED FOR THE INTERVAL ARITHMETIC I1 BENCHMARK

Solution	Variables values	Functions values	Solution	Variables values	Functions values
Sol. 1	0.0464905115	0.2077959240	Sol. 5	0.1169663983	0.1376466161
	0.1013568357	0.2769798846		-0.0360324410	0.4148982728
	0.0840577820	0.1876863212		-0.0517944631	0.3233536840
	-0.1388460309	0.3367887114		-0.1825907448	0.3816023122
	0.4943905739	0.0530391321		0.0741902056	0.3669485262
	-0.0760685163	0.2223730535		0.25036046290	0.1038035643
	0.2475819110	0.1816084752		0.2043019803	0.2250628007
	-0.0170748156	0.0874896386		0.0120607075	0.0595255950
	0.0003667535	0.3447200366		0.18799376080	0.1571104516
	0.1481119311	0.2784227489		0.09312965555	0.3333260708
Sol. 2	0.1224819761	0.1318552790	Sol. 6	0.0600624922	0.1941520526
	0.1826200685	0.1964428361		0.0665034453	0.3118493104
	0.2356779803	0.0364987069		0.1163378165	0.1553271371
	-0.0371150470	0.2354890155		-0.0456993775	0.2437021588
	0.3748181856	0.0675753064		0.1649150798	0.2765628684
	0.2213311341	0.0739986588		-0.1223771045	0.2690505556
	0.0697813035	0.3607038292		0.0666559953	0.3628858561
	0.0768058043	0.0059182979		0.0732866593	0.0028059669
	-0.0312153867	0.3767487763		0.0745961823	0.2703511071
	0.1452667120	0.2811693568		0.0578573421	0.3686104960
Sol. 3	0.0633944399	0.1908436653	Sol. 7	0.2077500302	0.0464943050
	0.1017426933	0.2767897367		0.0299198492	0.3489889696
	-0.1051842285	0.3769063436		-0.0339491324	0.3058418474
	-0.0477059943	0.2460900702		-0.2027950317	0.4012915513
	0.4149858326	0.0260337751		0.2131771707	0.2284027988
	0.1215195321	0.0256054760		0.0568458067	0.0886970244
	0.2539777159	0.1761486401		0.2267650517	0.2024745658
	0.0843972823	0.1349869851		-0.0977041236	0.1687259437
	-0.0534132992	0.3986395691		-0.0339921200	0.3787652675
	0.0880998746	0.3383563536		0.2532921324	0.1741025236
Sol. 4	0.1939820199	0.0603335280	Sol. 8	-0.0364260444	0.2907604740
	0.0152114400	0.3633514726		0.1232874096	0.2550909534
	0.1618654345	0.1097465792		-0.0349926786	0.3065546443
	0.0056985809	0.1914653768		0.0959206680	0.1020362156
	0.1904538879	0.2502358229		0.2474776135	0.1940393232
	-0.1623604033	0.3089460561		0.0877790534	0.0582777294
	0.1864448178	0.2428992222		0.2453311373	0.1832428336
	-0.0449302706	0.1144916285		-0.1234286095	0.1938589990
	0.1675935311	0.1774161896		-0.0767543100	0.4216253107
	-0.0274959004	0.4539962587		0.0837953112	0.3428082855

Then, the ith quasi-Newton step δx_i is the solution of

$$B_i \delta x i = -f_i$$

where

$$\delta x_i = x_{i+1} x_i.$$

The quasi-Newton or secant condition is that B_{i+1} satisfies

$$B_{i+1} \delta x_i = \delta f_i$$

where

$$\delta f_i = f_{i+1} - f_i.$$

This is the generalization of the 1-D secant approximation to the derivative $\delta f / \delta x$. Many different auxiliary conditions to pin down B_{i+1} have been explored; however, the best-performing algorithm in practice results from Broyden's formula. This formula is based on the idea of getting B_{i+1} by making the least change to B_i consistent with the secant equation. Broyden illustrated that the resulting formula is given by

$$B_{i+1} = B_i + \frac{(\delta f_i - B_i \delta x_i) \otimes \delta x_i}{(\delta x_i)^2}.$$

3) Secant Method: The secant method [40] is a root-finding algorithm that uses a succession of roots of secant lines to better approximate a root of a function. The secant method is defined by the recurrence relation

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n).$$

As evident from the recurrence relation, the secant method requires two initial values, i.e., x_0 and x_1, which should ideally be chosen to lie close to the root. Referring to Fig. 2, two points a and b are initially considered. Then, the secant of the chord of the graph of function f through the points $(a, f(a))$, $(b, f(b))$ is defined as

$$y - f(b) = \frac{f(b) - f(a)}{b - a}(x - b).$$

The point c is chosen to be the root of this line such that

$$f(b) + \frac{f(b) - f(a)}{b - a}(c - b) = 0.$$

Solving this equation gives the recurrence relation for the secant method. The new value c is equal to x_{n+1}, and b and a are x_n and x_{n-1}, respectively.

4) Effati and Nazemi Method: Effati and Nazemi [18] proposed a new method for solving systems of nonlinear equations. The method proposed in [18] is shortly presented below.

The following notations are used:

$$\begin{aligned} x_i(k+1) &= f_i\left(x_1(k), x_2(k), \ldots, x_n(k)\right) \\ f(x_k) &= \left(f_1(x_k), f_2(x_k), \ldots, f_n(x_k)\right) \\ i &= 1, 2 \ldots, n \quad \text{and} \quad x_i : N \to \Re. \end{aligned}$$

If there exists a t such that $x(t) = 0$, then $f_i(x(t-1)) = 0$, $i = 1, \ldots, n$. This involves that $x(t-1)$ is an exact solution for the given system of equations.

Let us define

$$\begin{aligned} u(k) &= \left(u_1(k), u_2(k), \ldots, u_n(k)\right) \\ x(k+1) &= u(k). \end{aligned}$$

Define $f^0 : \Omega \times U \to \Re$ (Ω and U are compact subsets of $\Re^n$) as follows:

$$f^0\left(x(k), u(k)\right) = \left\| u(k) - f\left(x(k)\right) \right\|_2^2.$$

The error function E is defined as follows:

$$\begin{aligned} E[x^t, u^t] &= \sum_{k=0}^{t-1} f^0\left(x(k), u(k)\right). \\ x^t &= \left(x(1), x(2), \ldots, x(t-1), 0\right) \\ u^t &= \left(u(1), u(2), \ldots, u(t-1), 0\right). \end{aligned}$$

Consider the following problem:

$$(P_1) \begin{cases} \text{minimize } E[x^t, u^t] = \sum_{k=0}^{t-1} f^0\left(x(k), u(k)\right) \\ \text{subject to} \\ x(k+1) = u(k) \\ x(0) = 0, \; x(t) = 0 \; (x^0 \text{ is known}). \end{cases}$$

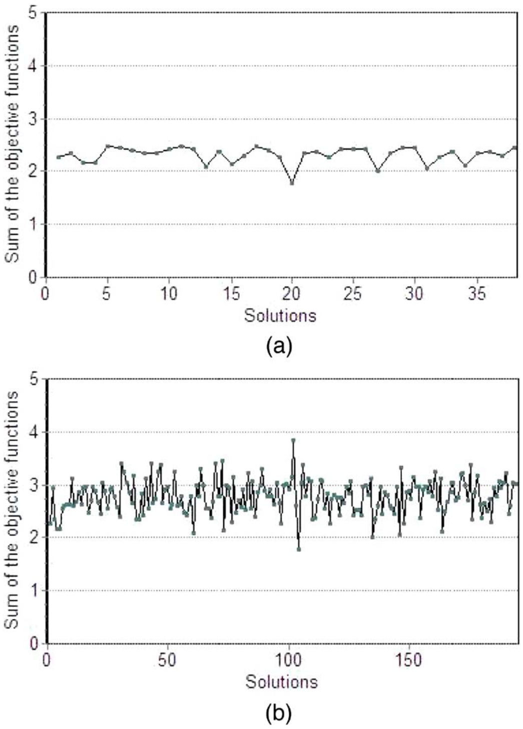

Fig. 6. Nondominated solutions obtained for i1 example. (a) Solutions whose sum of the absolute values of the objective functions is less than or equal to 2.5. (b) Sum of the absolute values of the objective functions for all the nondominated solutions obtained.

TABLE IX
PARAMETERS USED BY THE EVOLUTIONARY APPROACH FOR THE NEUROPHYSIOLOGY APPLICATION

Parameter	Value
Population size	300
External set size	200
Number of generations	200
Sigma (for mutation)	0.13
Tournament size	5

As illustrated in [18], if there is an optimal solution for the problem (P_1) such that the value of E will be zero, then this is also a solution (an exact solution) for the system of equations to be solved. The problem is transformed to a measure theory problem. By solving the transformed problem, u^t is firstly constructed, and from there, x^t is obtained. The reader is advised to consult [18] for more details. The measure theory method is improved in [18]. The interval $[1, t]$ is divided into the subintervals $S_1 = [1, t-1]$ and $S_2 = [t-1, t]$. Problem (P_1) is solved in both subintervals, and errors E_1 and E_2 are obtained, respectively. This way, an upper bound for the total error is found. If this upper bound is estimated to be zero, then an approximate solution for the problem is found.

III. TRANSFORMATION INTO A MULTIOBJECTIVE OPTIMIZATION PROBLEM

Some basic definitions of a multiobjective optimization problem and the optimality concept of the solutions [48] are presented in this section.

TABLE X
EXAMPLES OF NONDOMINATED SOLUTIONS OBTAINED FOR THE NEUROPHYSIOLOGY APPLICATION

Solution	Variables values	Functions values	Solution	Variables values	Functions values
Sol. 1	-0.8282192996	0.3139636069	Sol. 7	0.6609930931	0.0837968094
	0.5446434961	0.1206333341		-0.4821043312	0.0055056402
	-0.0094437659	0.0652332757		0.8042915766	0.3405628319
	0.7633676230	0.0123681793		-0.8729660781	0.2381471738
	0.0199325983	0.0465408323		-0.8987020407	0.3141261109
	0.1466452805	0.0330776356		-0.1909288931	0.2770687343
Sol. 2	-0.6512719807	0.3607740323	Sol. 8	0.2739187045	0.0075566488
	-0.6858609598	0.1134596029		0.1077541336	0.1344681018
	-0.4637572369	0.0143291397		0.9656734396	0.3079429049
	-0.6450853748	0.0412380343		0.9240784300	0.0065004171
	0.1535909562	0.0204607154		-0.3143660356	0.0831983095
	-0.0036883801	0.0290928705		-0.0314940456	0.0231155825
Sol. 3	0.0425943625	0.1489636110	Sol. 9	-0.0838634907	0.2102336348
	-0.1626952821	0.0049729625		-0.1437222650	0.2606391818
	-0.9215324786	0.3332320690		0.8847221485	0.0953305060
	0.9841530788	0.0038536711		0.8477645479	0.0000303505
	-0.6789794019	0.1183698936		0.1777227339	0.0069629557
	-0.9070329917	0.0224932754		-0.0455327341	0.0003085023
Sol. 4	0.3269911198	0.2710537507	Sol. 10	0.4612359064	0.0889325997
	0.0266425162	0.0257807695		0.2783584687	0.1108731832
	0.7886843835	0.1331679003		0.9360523694	0.0852199228
	0.9866658030	0.0083976429		-0.9009125260	0.0198955321
	-0.2403284017	0.0421023524		-0.1421082154	0.1197727508
	0.2613854687	0.2008350659		-0.2759388163	0.0090365095
Sol. 5	0.8625703877	0.2043564483	Sol. 11	-0.6907741758	0.2281557150
	-0.7176375053	0.2177684569		0.8565963646	0.0325274349
	-0.2271912801	0.0227051752		-0.5428400528	0.0273082414
	0.5169409578	0.1404211857		0.5465986672	0.0385212104
	-0.1305290129	0.0352067233		-0.2327716625	0.0318257635
	0.1532817352	0.0628718325		-0.0607828078	0.0359158033
Sol. 6	0.7618711576	0.1608754444	Sol. 12	-0.8078668904	0.0050092197
	0.6775336796	0.0901846503		-0.9560562726	0.0366973076
	0.5086028850	0.0169111046		0.5850998782	0.0124852708
	-0.6713892035	0.1306483985		-0.2219439027	0.0276342907
	0.2563543063	0.0674916558		0.0620152964	0.0168784849
	0.0555642759	0.0585551925		-0.0057942792	0.0248569233

Let Ω be the search space. Consider n objective functions f_1, $f_2, \ldots, f_n$, i.e.,

$$f_i : \Omega \rightarrow \Re, \qquad i = 1, 2, \ldots, n$$

where $\Omega \subset \Re^m$.

The multiobjective optimization problem is defined as

$$\begin{cases} \text{optimize } f(x) = (f_1(x), \ldots, f_n(x)) \\ \text{subject to } x = (x_1, x_2, \ldots x_m) \in \Omega. \end{cases}$$

For deciding whether a solution is better than another solution or not, the following relationship between solutions might be used.

Definition 1—Pareto Dominance: Consider a maximization problem. Let x and y be two decision vectors (solutions) from Ω.

Solution x dominates y (also written as $x \succ y$) if and only if the following conditions are fulfilled.

1) $f_i(x) \geq f_i(y), \forall i = 1, 2, \ldots, n.$
2) $\exists j \in \{1, 2, \ldots, n\}$: $f_j(x) > f_j(y).$

That is, a feasible vector x is Pareto optimal if no feasible vector y can increase some criterion without causing a simultaneous decrease in at least one other criterion. In the literature, other terms have also been used instead of the Pareto optimal or minimal solutions, including words such as nondominated, noninferior, efficient, and functional-efficient solutions. The solution x^0 is *ideal* if all objectives have their optimum in a common point x^0.

Definition 2—Pareto Front: The images of the Pareto optimum points in the criterion space are called the *Pareto front*. The system of equations (P) can be transformed into a multiobjective optimization problem. Each equation can be considered as an objective function. The goal of this optimization function is to minimize the difference (in absolute value) between the left side and the right side of the equation. Since the right term is zero, the objective function is to be given by the absolute value of the left term.

The system (P) is then equivalent to

$$(P)\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0 \\ f_2(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

$$\Leftrightarrow (P')\begin{cases} \text{minimize abs}\,(f_1(x_1, x_2, \ldots, x_n)) \\ \text{minimize abs}\,(f_2(x_1, x_2, \ldots, x_n)) \\ \vdots \\ \text{minimize abs}\,(f_n(x_1, x_2, \ldots, x_n)). \end{cases}$$

IV. Evolutionary Nonlinear Equations System

Evolutionary algorithms are ubiquitous nowadays, having been successfully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, operations research, bioinformatics, and social systems. In many cases, the mathematical function that describes the problem is not known, and the values at certain parameters are obtained from simulations. In contrast to many other optimization techniques, an important advantage of evolutionary algorithms is that they can cope with multimodal functions.

An evolutionary algorithm approach is proposed for solving the multiobjective optimization problem obtained by transforming the system of equations. The following steps may be used.

Initialization: Some starting points (initial solutions) are generated based on the problem domain of definition (which can be approximated for each particular problem). A real representation of solution is considered. Each solution is a vector whose length is equal to the number of variables for the considered system of equations.

Evolution Phase: The current available solution is evolved in an iterative manner. Genetic operators (such as crossover and mutation) are used. By applying crossover between two solutions, two new solutions are obtained, which are convex combinations of the initial two solutions. The mutation operator produces modification of the solution over which it is applied. This consists of generating a new solution with Gaussian distribution (or any other distribution) starting from the initial considered solution (the one that is mutated). More details about several existing forms of these operators can be found in [1], [19], [20], and [45]–[47].

To compare two solutions, the Pareto dominance relationship is used. An external set is used where all the nondominated solutions found during the iteration process are stored. The size of this external set is fixed and depends on the number of nondominated solutions to be obtained at the end of the search process. At each iteration, this set is updated by introducing all the nondominated solutions obtained at the respective step and by removing from the external set all solutions that will become dominated. When the size of this set is overloaded, some of the solutions are removed.

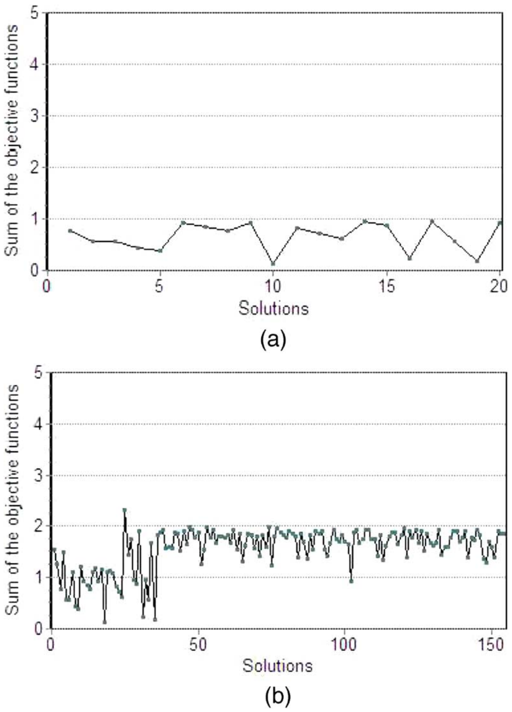

Fig. 7. Nondominated solutions obtained for the neurophysiology application. (a) Solutions whose sum of the absolute values of the objective functions is less than or equal to 1. (b) Sum of the absolute values of the objective functions for all the nondominated solutions obtained.

TABLE XI
Parameters Used by the Evolutionary Approach for the Chemical Equilibrium Application

Parameter	Value
Population size	500
External set size	200
Number of generations	500
Sigma (for mutation)	0.3
Tournament size	5

There are several criteria that must be taken into account while comparing two nondominated solutions in order to select one of them. One of the conditions that can be used in our case for comparing solutions that are nondominated is to consider as being the best solution among the two solutions the one for which the sum of all absolute values of the objectives is closer to zero. This means that, overall, the value of each objective is close to zero, or there is good balance between objectives having a lower value (desired) and objectives having a higher value (undesired). In other words, let us consider two n-dimensional solutions x and y (which means that the considered system is having n variables and n equations) and the equivalent system of equations as given by P'. One way to compare the solutions is to consider that the solution x is better than the solution y if

$$\sum_{i=1}^{n} |f_i(x)| < \sum_{i=1}^{n} |f_i(y)|.$$

The aforementioned principle is used to select which solutions to be kept into the external set for the next iteration as well

TABLE XII
EXAMPLES OF NONDOMINATED SOLUTIONS OBTAINED BY THE EVOLUTIONARY APPROACH FOR THE CHEMICAL EQUILIBRIUM APPLICATION

Solution	Variables values	Functions values	Solution	Variables values	Functions values
Sol. 1	-0.0163087455	0.1525772447	Sol. 8	0.1296088399	0.0660183067
	0.2613604709	0.3712483549		0.2275206857	0.1191351931
	0.5981559224	0.0265535280		-0.1061140447	0.2387977164
	0.8606983883	0.2784694066		-0.8124975178	0.0792417966
	0.0440020125	0.1168649339		0.0310264084	0.1760779901
Sol. 2	0.3357311285	0.3491354953	Sol. 9	0.3367208030	0.2796624109
	0.1015972384	0.3388481591		0.1420207287	0.0856442848
	0.1959807715	0.0324919199		-0.1427721429	0.2660909007
	0.5298149584	0.2853430985		-0.8435618534	0.0247927741
	0.0069016628	0.3380473798		0.0349599086	0.1028940828
Sol. 3	0.3273318676	0.3206895328	Sol. 10	0.4224008806	0.2712921597
	0.0396552907	0.2986524101		-0.1889079092	0.0011063024
	0.5208586308	0.0741335715		0.3561384679	0.1890226787
	-0.4442729860	0.1331193703		-0.7390069308	0.1414158627
	0.0065409250	0.3989667326		0.0237712845	0.1106294577
Sol. 4	0.1626252165	0.0298611415	Sol. 11	0.2395706253	0.2975483261
	-0.3017126041	0.2240794988		0.4637567881	0.2847994096
	-0.1783889066	0.2301576033		0.0154833612	0.1413409742
	0.7137275251	0.0971666758		-0.5861394961	0.02122094995
	0.0278993324	0.3805446921		0.0177082676	0.3056084889
Sol. 5	0.3221051215	0.3510711865	Sol. 12	0.1671662105	0.1245889560
	0.2525051386	0.3544681189		0.2358035522	0.0243717563
	-0.4189363818	0.0165037731		-0.1121274369	0.2079331799
	-0.6331604264	0.1035503586		-0.7953186041	0.1717817639
	0.0174557111	0.1177149436		0.0273318802	0.1555592488
Sol. 6	0.0429158354	0.0446360606	Sol. 13	0.4586478321	0.3386346204
	0.0811055468	0.2478304056		-0.1011456067	0.1204471133
	-0.2662203512	0.2040211938		-0.0115635220	0.1963075419
	-0.7711670069	0.0243731377		-0.75891034603	0.1702524624
	0.0303442027	0.3396013173		0.02454099949	0.0117867162
Sol. 7	0.7276812579	0.3738656386	Sol. 14	0.4064810686	0.3392512646
	-0.4551167619	0.0304080605		0.0073246701	0.1784649874
	-0.2113909448	0.0838221369		0.0846953560	0.1843099545
	-0.4895999565	0.1776147414		0.6726843741	0.0310865463
	0.0075452271	0.3755303113		0.0234023812	0.1365650475

as whenever we wish to choose between two nondominated solutions (for instance, after applying crossover while choosing between parents and offspring and/or after applying mutation). Several other criteria can be taken into account. For instance, we wish to obtain a very low value (very close to zero) for some of the objectives, and then we can accordingly choose our solution. However, this requires detailed knowledge about the objectives.

We consider as the current population of the next iteration the population obtained by unifying the current population of the previous iteration and the external set. The main steps of the evolutionary approach used are presented in Algorithm 2. The termination criteria of Algorithm 2 refer to a specified number of iterations.

Algorithm 2 The iterative evolutionary steps proposed for solving nonlinear equations systems.

```
Step 1.
    Set $t = 0$.
    Randomly generate starting solutions $P(t)$ on a given domain.
    Select all the nondominated solutions from $P(t)$ and store them into the external set $E$ containing the nondominated solutions found so far.
    If the cardinal of $E$ exceeds the maximum allowed size, reduce the number of solutions with respect to the sum of the absolute values of the objectives.
Step 2.
    Step 2.1. Apply crossover (with a given probability) on $P(t) \cup E$ until a number of new individuals equal to the size of $P(t)$ are obtained.
    Let $Q(t)$ be the set obtained from the best between the solutions that are combined and the solutions obtained after recombination (Pareto domination relation is applied).
    Step 2.2. Mutate (with a given probability) all the individuals from $Q(t)$.
```

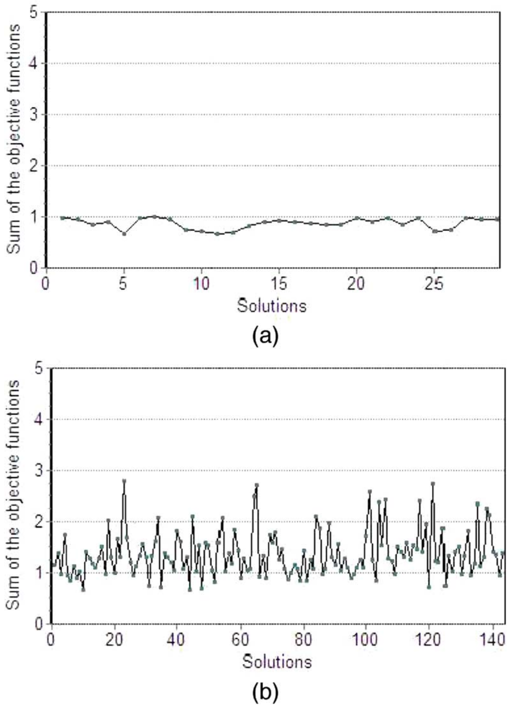

Fig. 8. Nondominated solutions obtained for the chemical equilibrium application. (a) Solutions whose sum of the absolute values of the objective functions is less than or equal to 1. (b) Sum of the absolute values of the objective functions for all the nondominated solutions obtained.

```
    Step 2.3. Update $E$ with the nondominated individuals
    from $P(t) \bigcup Q(t)$ and apply the reduction procedure if the
    allowed size of $E$ is exceeded.
    Step 2.4. Set $t = t + 1$.
        $P(t) = Q(t)$.
Step 3.
    If the termination criteria are reached, go to step 4.
    Otherwise, go to step 2.
Step 4.
    Print $E$.
```

V. Experiments and Results

This section reports several experiments and comparisons using the proposed approach. Some well-known applications are also considered in the subsequent section.

A. Illustrative Example

In this example, a simple equations system (two equations) is used, which is having more than one solution, and the ability of the proposed approach is illustrated to detect several solutions in a single run. The following equations system is considered:

$$\begin{cases} x^2 - y^2 = 0 \\ 1 - |x - y| = 0. \end{cases}$$

The two functions corresponding to the transformed problem into a multiobjective optimization problem are depicted in Fig. 3.

TABLE XIII
COEFFICIENTS a_{ki} FOR THE KINEMATIC EXAMPLE KIN2

-0.249150680	+0.125016350	-0.635550077	+1.48947730
+1.609135400	-0.686607360	-0.115719920	+0.23062341
+0.279423430	-0.119228120	-0.666404480	+1.32810730
+1.434801600	-0.719940470	+0.110362110	-0.25864503
+0.000000000	-0.432419270	+0.290702030	+1.16517200
+0.400263840	+0.000000000	+1.258776700	-0.26908494
-0.800527680	+0.000000000	-0.629388360	+0.53816987
+0.000000000	-0.864838550	+0.581404060	+0.58258598
+0.074052388	-0.037157270	+0.195946620	-0.20816985
-0.083050031	+0.035436896	-1.228034200	+2.68683200
-0.386159610	+0.085383482	+0.000000000	-0.69910317
-0.755266030	+0.000000000	-0.079034221	+0.35744413
+0.504201680	-0.039251967	+0.026387877	+1.24991170
-1.091628700	+0.000000000	-0.057131430	+1.46773600
+0.000000000	-0.432419270	-1.162808100	+1.16517200
+0.049207290	+0.000000000	+1.258776700	+1.07633970
+0.049207290	+0.013873010	+2.162575000	-0.69686809

TABLE XIV
PARAMETERS USED BY THE EVOLUTIONARY APPROACH FOR THE KINEMATIC APPLICATION

Parameter	Value
Population size	500
External set size	200
Number of generations	1000
Sigma (for mutation)	0.5
Tournament size	5

The parameter values used by the evolutionary approach are given in Table I.

After applying the evolutionary approach, several nondominated solutions are obtained. Some of the solutions are presented in Table II, and the Pareto curve is depicted in Fig. 4.

The sum of the absolute values of the objectives is plotted in Fig. 5.

B. Numerical Comparisons

1) Two Equations Systems: We considered the same problems (Examples 1 and 2) used by Effati and Nazemi [18]. The algorithm proposed by Effati and Nazemi is compared with Newton's method, the secant method, and Broyden's method. Only systems of two equations were considered by Effati and Nazemi. The parameters used by the evolutionary approach for Examples 1 and 2 are given in Table III.

Example 1: Consider the following nonlinear system:

$$\begin{cases} f_1(x_1, x_2) = \cos(2x_1) - \cos(2x_2) - 0.4 = 0 \\ f_2(x_1, x_2) = 2(x_2 - x_1) + \sin(2x_2) - \sin(2x_1) - 1.2 = 0. \end{cases}$$

The results obtained by applying Newton, secant, Broyden, and Effati methods and the proposed method are presented in Table IV.

Example 2: The following system is considered:

$$\begin{cases} f_1(x_1, x_2) = e^{x_1} + x_1 x_2 - 1 = 0 \\ f_2(x_1, x_2) = \sin(x_1 x_2) + x_1 + x_2 - 1 = 0. \end{cases}$$

The results obtained by Effati and Nazemi's method and by the evolutionary approach are given in Table V.

TABLE XV
EXAMPLES OF NONDOMINATED SOLUTIONS OBTAINED FOR THE KINEMATIC APPLICATION KIN2

Solution	Variables values	Functions values	Solution	Variables values	Functions values
Sol. 1	-0.0625820337	0.3911967824	Sol. 6	-0.3064809352	0.9060378884
	0.7777446281	0.3925758963		-0.0056167467	0.7492384561
	-0.0503725828	0.8526542737		-0.5007294639	0.7403485971
	0.3805368959	0.5424213097		-0.0944531990	0.4782413752
	-0.5592587603	0.7742116224		-0.7161265376	0.8882665877
	-0.6988338865	0.1537105718		0.6371096100	0.2081000785
	0.3963927675	0.9116019977		0.4792262814	0.7690562864
	0.0861763643	0.1519175234		-0.4680908930	0.4979341279
Sol. 2	-0.1564353525	0.7723864643	Sol. 7	-0.5739275815	0.5286935399
	0.4507122320	0.5832167209		0.3767142036	0.8412047222
	0.4622139796	0.0087255337		0.1299295442	0.3390733979
	-0.8818348503	0.2031050697		-0.8025240903	0.2643422170
	-0.6522824284	0.6056929403		-0.3026761756	0.5162506493
	0.4082826235	0.3663682493		0.1226779594	0.2659449070
	0.4718261386	0.3532359802		0.7855891471	0.4029986127
	-0.5070478474	0.4646334692		-0.1240877403	0.5347985046
Sol. 3	-0.5618177814	0.0336943748	Sol. 8	0.7822939914	0.1443884130
	0.8473813517	0.2323541264		-0.4935865657	0.1116189082
	0.2226897354	0.9432510244		0.8029653752	0.3487762429
	-0.0846064846	0.3663913629		0.0804385674	0.1428116670
	-0.7914861841	0.0081327333		0.9223437373	0.3802921343
	-0.4111014166	0.2206688453		0.0296919251	0.5311414488
	0.3056098314	0.624967041		0.7980078255	0.0338066582
	0.4290046184	0.4690397335		-0.8324923146	0.0083023855
Sol. 4	0.0608363294	0.7409305497	Sol. 9	-0.7461742647	0.0762392460
	0.5053398770	0.6356638946		0.6057926381	0.2265885749
	0.3301025811	0.7726033233		0.9271482376	0.0694226055
	-0.3441350935	0.8133740699		0.2664085959	0.6991957509
	-0.2611454909	0.0176109130		-0.4794066217	0.0290725466
	-0.1335439441	0.0929407653		-0.3302191010	0.1921690687
	0.7518856650	0.2565663633		0.7692157072	0.4459273853
	-0.3711959678	0.260158013		-0.1642217317	0.7592540224
Sol. 5	-0.7461742647	0.1552859413	Sol. 10	-0.6205399028	0.1461818771
	0.5365985698	0.4712662401		0.6846519932	0.4555568728
	0.4907094198	0.7461140527		-0.9933817672	0.0102591212
	0.1144124666	0.6917033188		-0.0541621937	0.9914691316
	-0.5433290610	0.9909917470		-0.0748152730	0.2920414356
	0.3443841680	0.0271473496		0.37913950333	0.5909270772
	0.6758924483	0.0900355495		-0.4826393335	0.8366810215
	-0.3341809018	0.20247922410		-0.7830381952	0.7918662138

C. Systems Having More Than Two Equations

As evident from the above experiment results, the proposed approach has obtained better results as compared to the other established techniques. It is to be noted that Effati and Nazemi's technique was only applied for two equations systems. Since our approach was promising as compared to Effati and Nazemi's technique (which, in turn, was outperforming some other standard known techniques), we decided to extend our approach for more complex equations systems. The examples considered in the following section are listed in Table VI and are well-known difficult benchmarks for numerical analysis researchers.

1) Interval Arithmetic Benchmark: We consider one benchmark problem proposed from interval arithmetic [29], [33] (see also [27]). The benchmark consists of the following system of equations:

$$\begin{cases} 0 = x_1 - 0.25428722 - 0.18324757 x_4 x_3 x_9 \\ 0 = x_2 - 0.37842197 - 0.16275449 x_1 x_{10} x_6 \\ 0 = x_3 - 0.27162577 - 0.16955071 x_1 x_2 x_{10} \\ 0 = x_4 - 0.19807914 - 0.15585316 x_7 x_1 x_6 \\ 0 = x_5 - 0.44166728 - 0.19950920 x_7 x_6 x_3 \\ 0 = x_6 - 0.14654113 - 0.18922793 x_8 x_5 x_{10} \\ 0 = x_7 - 0.42937161 - 0.21180486 x_2 x_5 x_8 \\ 0 = x_8 - 0.07056438 - 0.17081208 x_1 x_7 x_6 \\ 0 = x_9 - 0.34504906 - 0.19612740 x_{10} x_6 x_8 \\ 0 = x_{10} - 0.42651102 - 0.21466544 x_4 x_8 x_1. \end{cases}$$

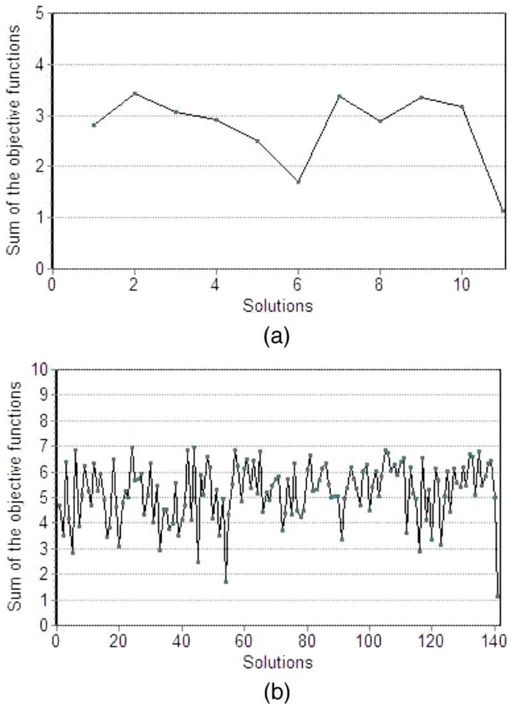

Fig. 9. Nondominated solutions obtained for the kinematic application kin2. (a) Solutions whose sum of the absolute values of the objective functions is less than or equal to 3.5. (b) Sum of the absolute values of the objective functions for all the nondominated solutions obtained.

Parameters used by the evolutionary approach are listed in Table VII. Some of the nondominated solutions obtained as well as the function values (which represent the values of the system's equations obtained by replacing the variable values) are presented in Table VIII. The sum of the absolute values of the objectives for the nondominated solutions obtained is depicted in Fig. 6. In Fig. 6(a), the nondominated solutions for which the sum of objectives (in absolute values) is less than or equal to 2.5 are plotted. In Fig. 6(b), the sum of objectives (in absolute values) for all the nondominated solutions obtained is depicted.

2) Neurophysiology Application: We considered the example proposed in [50], which consisted of the following equations:

$$\begin{cases} x_1^2 + x_3^2 = 1 \\ x_2^2 + x_4^2 = 1 \\ x_5 x_3^3 + x_6 x_4^3 = c_1 \\ x_5 x_1^3 + x_6 x_2^3 = c_2 \\ x_5 x_1 x_3^2 + x_6 x_4^2 x_2 = c_3 \\ x_5 x_1^2 x_3 + x_6 x_2^2 x_4 = c_4. \end{cases}$$

The constants c_i can be randomly chosen. In our experiments, we considered $c_i = 0, i = 1, \ldots, 4$. In [27], this problem is used to show the limitations of Newton's method for which the running time exponentially increases with the size of the initial intervals. We considered the following values for the parameters used by the evolutionary approach as given in Table IX.

TABLE XVI
PARAMETERS USED BY THE EVOLUTIONARY APPROACH FOR THE COMBUSTION APPLICATION

Parameter	Value
Population size	500
External set size	200
Number of generations	300
Sigma (for mutation)	0.1
Tournament size	5

Some of the nondominated solutions obtained by our approach as well as the values of the objective functions for these values are presented in Table X. The sum of the absolute values of the objectives for the nondominated solutions obtained is depicted in Fig. 7. In Fig. 7(a), the nondominated solutions for which the sum of objectives (in absolute values) is less than or equal to 1 are plotted. In Fig. 7(b), the sum of the objectives (in absolute values) for all the nondominated solutions obtained is depicted.

3) Chemical Equilibrium Application: We consider the chemical equilibrium system as given by the following [32] (see also [27]):

$$\begin{cases} x_1 x_2 + x_1 - 3x_5 = 0 \\ 2x_1 x_2 + x_1 + x_2 x_3^2 + R_8 x_2 - R x_5 \\ \quad +2R_{10} x_2^2 + R_7 x_2 x_3 + R_9 x_2 x_4 = 0 \\ 2x_2 x_3^2 + 2R_5 x_3^2 - 8x_5 + R_6 x_3 + R_7 x_2 x_3 = 0 \\ R_9 x_2 x4 + 2x_4^2 - 4R x_5 = 0 \\ x_1(x_2 + 1) + R_{10} x_2^2 + x_2 x_3^2 + R_8 x_2 \\ \quad +R_5 x_3^2 + x_4^2 - 1 + R_6 x_3 + R_7 x_2 x_3 + R_9 x_2 x_4 = 0 \end{cases}$$

where

$$\begin{cases} R = 10 \\ R_5 = 0.193 \\ R_6 = \frac{0.002597}{\sqrt{40}} \\ R_7 = \frac{0.003448}{\sqrt{40}} \\ R_8 = \frac{0.00001799}{40} \\ R_9 = \frac{0.0002155}{\sqrt{40}} \\ R_{10} = \frac{0.00003846}{40}. \end{cases}$$

The parameters used by the evolutionary approach are presented in Table XI. Some of the nondominated solutions obtained by the evolutionary approach for the chemical equilibrium application are depicted in Table XII.

The sum of the absolute values of the objectives for the nondominated solutions obtained is depicted in Fig. 8. In Fig. 8(a), the nondominated solutions for which the sum of objectives (in absolute values) is less than or equal to 1 are plotted. In Fig. 8(b), the sum of the objectives (in absolute values) for all the nondominated solutions obtained is depicted.

4) Kinematic Application: We consider the kinematic application kin2 as introduced in [34] (see also [27]), which describes the inverse position problem for a six-revolute-joint

TABLE XVII
EXAMPLES OF NONDOMINATED SOLUTIONS OBTAINED FOR THE COMBUSTION APPLICATION

Solution	Variables values	Functions values	Solution	Variables values	Functions values
Sol. 1	-0.0552429896	0.0274133878	Sol. 5	0.0348357700	0.0289865877
	-0.0023377533	0.0841848522		0.1092386108	0.2167589209
	0.0455880930	0.1482418893		0.1250085306	0.0089588165
	-0.1287029472	0.0839188567		-0.0107958218	0.1683941428
	0.0539771728	0.0030517851		-0.1399310251	0.0012135380
	-0.0151036079	0.0000109317		-0.0177642059	0.0238661499
	0.1063159019	0.0165644486		-0.0787941605	0.0001165497
	0.0386267592	0.0025184283		0.0917803902	0.0043547546
	-0.1144905135	0.0001291515		-0.0422169082	0.0038054137
	0.0872294353	0.0000003019		-0.0302349392	0.0004156978
Sol. 2	-0.0338378558	0.0008794626	Sol. 6	0.0172348545	0.0202656999
	0.0185669333	0.1035086837		-0.0049839785	0.0113645412
	0.0534924988	0.0955626197		-0.0036835674	0.0052992119
	-0.0392784517	0.2441423777		-0.0401647761	0.0609700462
	0.0183882247	0.0011449995		0.0334303826	0.0002970384
	0.0005245892	0.0006894619		-0.0049589041	0.0000496805
	-0.1024269629	0.0015427967		0.0505724112	0.0016132092
	0.0500461848	0.0018100789		-0.0076509738	0.0000634846
	-0.1013361102	0.0006282589		-0.1141678724	0.0000858910
	0.0404252678	0.0000116649		0.0544069796	0.0000004281
Sol. 3	-0.0103209333	0.1170689295	Sol. 7	-0.1443475355	0.0460704291
	0.0021201108	0.0726549501		0.0137124749	0.0928317803
	0.1207182825	0.2023013696		0.0532523778	0.0470659155
	-0.0263026679	0.0155834694		-0.0407593315	0.0014507164
	0.0044219824	0.0001065214		0.0053340672	0.0208362107
	-0.0850838579	0.0000089983		0.0390841261	0.0003760600
	0.0053645992	0.0006918303		0.0196593075	0.0016613231
	-0.0480333324	0.0012459181		0.0396094025	0.0076868554
	0.0732269061	0.0000218860		0.0463257617	0.0019793648
	0.1059498141	0.0000000463		-0.0921334590	0.0000271419
Sol. 4	0.0177198747	0.0262522791	Sol. 8	0.1612054472	0.0966899590
	0.0030100424	0.0644480331		0.1001108591	0.0844303598
	0.0676669725	0.1379528266		-0.0303525758	0.0520483724
	-0.0408039903	0.286269853		0.0015541591	0.0511277742
	0.0852565598	0.0003139895		0.0464169709	0.0259871938
	0.0536056660	0.0000181153		0.0906816701	0.0200443591
	0.1635419218	0.0016649656		-0.0263359667	0.0000024154
	-0.0031889394	0.0011990507		-0.0540477839	0.0048929924
	-0.1390794276	0.0000533461		0.0577884947	0.0161384122
	0.0275601661	0.0000001605		-0.121281367	0.0016156306

problem in mechanics. The equations describe a denser constraint system and are given as follows:

$$\begin{cases} x_i^2 + x_{i+1}^2 - 1 = 0 \\ a_{1i}x_1x_3 + a_{2i}x_1x_4 + a_{3i}x_2x_3 + a_{4i}x_2x_4 + \\ a_{5i}x_2x_7 + a_{6i}x_5x_8 + a_{7i}x_6x_7 + a_{8i}x_6x_8 + \\ a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 + a_{13i}x_5 + a_{14i}x_6 + \\ a_{15i}x_7 + a_{16i}x_8 + a_{17i} = 0 \\ 1 \le i \le 4. \end{cases}$$

The coefficients a_{ki}, $1 \le k \le 17$, $1 \le i \le 4$, are given in Table XIII.

The parameters used by the evolutionary approach for the kinematic example kin2 are presented in Table XIV. Some of the nondominated solutions obtained by the evolutionary approach for the kinematic example kin2 are presented in Table XV.

The sum of the absolute values of the objectives for the nondominated solutions obtained for the kinematic application kin2 is depicted in Fig. 9. In Fig. 9(a), the nondominated solutions for which the sum of objectives (in absolute values) is less than or equal to 1 are plotted. In Fig. 9(b), the sum of the objectives (in absolute values) for all the nondominated solutions obtained is depicted.

5) Combustion Application: We consider the combustion problem for a temperature of 3000 °C as proposed in [35] (see also [27]). The problem is described by the following sparse system of equations:

$$\begin{cases} x_2 + 2x_6 + x_9 + 2x_{10} = 10^{-5} \\ x_3 + x_8 = 3 \cdot 10^{-5} \\ x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} = 5 \cdot 10^{-5} \\ x_4 + 2x_7 = 10^{-5} \\ 0.5140437 \cdot 10^{-7} x_5 = x_1^2 \\ 0.1006932 \cdot 10^{-6} x_6 = 2x_2^2 \\ 0.7816278 \cdot 10^{-15} x_7 = x_4^2 \\ 0.1496236 \cdot 10^{-6} x_8 = x_1x_3 \\ 0.6194411 \cdot 10^{-7} x_9 = x_1x_2 \\ 0.2089296 \cdot 10^{-14} x_{10} = x_1x_2^2. \end{cases}$$

The parameters used by the evolutionary approach for the combustion application are presented in Table XVI. Some of the nondominated solutions obtained by the evolutionary approach are presented in Table XVII.

The sum of the absolute values of the objectives for the nondominated solutions obtained for the combustion application is depicted in Fig. 10. In Fig. 10(a), the nondominated solutions for which the sum of objectives (in absolute values) is less than or equal to 1 are plotted. In Fig. 10(b), the sum of the objectives (in absolute values) for all the nondominated solutions obtained is depicted.

6) Economics Modeling Application: The following modeling problem is considered as difficult and can be scaled up to arbitrary dimensions [35]. The problem is given by the following system of equations:

$$\begin{cases} \left(x_k + \sum\limits_{i=1}^{n-k-1} x_i x_{i+k}\right) x_n - c_k = 0, \quad 1 \le k \le n-1 \\ \sum\limits_{l=1}^{n-1} x_l + 1 = 0. \end{cases}$$

The constants c_k can be randomly chosen. We considered the value 0 for the constants in our experiments and the case of 20 equations.

The parameters used by the evolutionary approach for the combustion application are presented in Table XVIII. Some of the nondominated solutions obtained by the evolutionary approach are presented in Table XIX.

The sum of the absolute values of the objectives for the nondominated solutions obtained for the combustion application is depicted in Fig. 10. In Fig. 11(a), the nondominated solutions for which the sum of objectives (in absolute values) is less than or equal to 1.5 are plotted. In Fig. 11(b), the sum of the objectives (in absolute values) for all the nondominated solutions obtained is depicted.

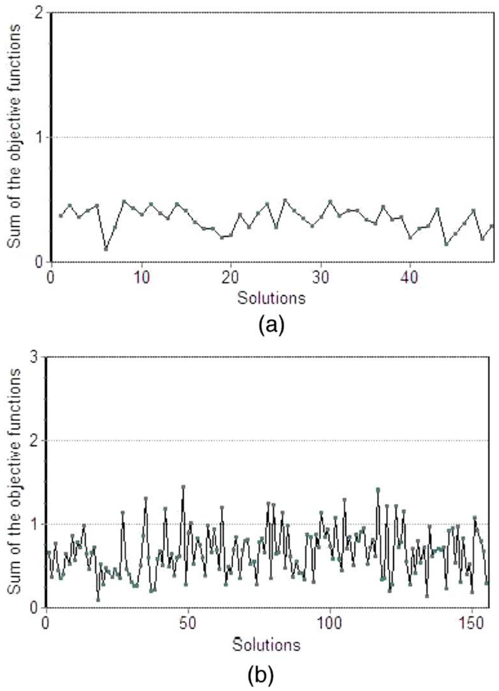

Fig. 10. Nondominated solutions obtained for the combustion application. (a) Solutions whose sum of the absolute values of the objective functions is less than or equal to 0.5. (b) Sum of the absolute values of the objective functions for all the nondominated solutions obtained.

TABLE XVIII
PARAMETERS USED BY THE EVOLUTIONARY APPROACH FOR THE ECONOMICS APPLICATION

Parameter	Value
Population size	500
External set size	200
Number of generations	300
Sigma (for mutation)	0.1
Tournament size	5

VI. DISCUSSIONS AND CONCLUSION

The proposed approach seems to be very efficient for solving equations systems. We analyzed the case of nonlinear equations systems. We first compared our approach for some simple equations systems having only two equations that were recently used for analyzing the performance of a new proposed method. The results obtained using the proposed evolutionary multiobjective optimization approach are very promising, clearly outperforming the new technique proposed by Effati and Nazemi and some of the classical methods established in the literature, namely, Newton, Broyden, and secant methods.

The running time required for our algorithm to converge is presented in Table XX. It is measured in seconds, and the experiments were run on a 2.4-GHz Intel Duo Core CPU with 2-GB RAM.

It is worth to mention that our approach obtains several nondominated solutions in one run (this number was kept constant, less than or equal to 200). Most of the systems used in the experiments have more than one solution (even hundreds of solutions can be found). Therefore, our approach detects multiple solutions in one run. If we consider that Newton-like methods obtain a single solution in 0.5 s, then 200 solutions will be obtained in 100 s. Also, these solutions are compared after the final run, and not all of them will be kept as final solutions. This means that more than 200 runs must be performed to obtain 200 solutions. For Examples 1 and 2, we are not aware of the running time required by Effati and Nazemi's algorithm. However, in Effati and Nazemi's approach, the search space is divided into 10 000 and 140 000 subsets, respectively, and it is obvious that it cannot be done in a few seconds.

The promising results obtained by our approach for two-equation systems were the starting point, and the approach was extended for high-dimensional nonlinear equations systems. We also used some of the most well known applications such as application from interval arithmetic benchmarks, application from neuropsychology, chemical equilibrium application, kinematic application, combustion application, and economics modeling. All these applications consist of systems having

TABLE XIX
EXAMPLES OF SOLUTIONS OBTAINED BY THE EVOLUTIONARY APPROACH FOR THE ECONOMICS MODELING APPLICATION e2

Solution	Variables values	Functions values	Solution	Variables values	Functions values
Sol. 1	-0.1639324	1.94E-5	Sol. 3	0.0936580	0.0004909
	-0.3813209	9.73E-5		-0.1113572	1.66E-5
	0.2242448	0.0001201		-0.0652960	0.0001413
	-0.0755094	2.39E-5		-0.0274100	0.0001387
	0.1171098	5.61E-5		-0.0515078	1.50E-5
	0.0174083	3.89E-5		-0.0525712	0.0001687
	-0.0594358	3.90E-5		0.1674281	0.0006289
	-0.2218284	9.31E-5		-0.0284058	7.11E-6
	0.1856304	0.0001294		-0.1587341	0.0002428
	-0.2653962	5.01E-5		-0.1284569	0.0001981
	-0.3712114	0.0001009		-0.1326800	0.0002817
	-0.3440810	0.0001601		-0.1138290	0.0002442
	-0.1060168	6.32E-6		-0.1430544	0.0001696
	0.0218564	7.96E-6		0.0521726	0.0001335
	-0.2028748	7.66E-5		-0.2608338	0.0005227
	0.0533728	2.35E-5		-0.1602811	0.0003038
	-0.0587111	2.21E-5		-0.1141750	0.0002455
	0.0057098	3.34E-6		-0.1677992	0.0003751
	-0.0149290	6.12E-6		-0.1159721	0.0002434
	-0.0004102	0.4110599		0.0020995	0.3975329
Sol. 2	-0.2071340	0.0050496	Sol. 4	-0.2686292	0.0015673
	-0.2251718	0.0003158		0.3391340	0.0028117
	-0.0910972	0.0035983		0.0732562	2.58E-5
	-0.0028412	0.0044357		0.0797120	0.0013986
	-0.2110337	0.0117217		-0.1109362	0.0003991
	0.4501557	0.0153466		0.0177894	0.0014187
	-0.0263800	0.0005013		0.4220681	0.0023095
	0.0086212	0.0001027		-0.0583526	4.67E-6
	-0.2065700	0.0076698		-0.2610232	0.0017655
	0.1663536	0.0045637		-0.2838340	0.0014891
	-0.1450036	0.0009652		-0.3579828	0.0028633
	-0.0743482	8.36E-6		0.0214270	0.0014185
	-0.2007066	0.0018678		-0.6282558	0.0035229
	-0.1451752	0.0016314		0.2185146	0.0010139
	-0.2078702	0.0044618		-0.0897853	0.0008279
	-0.2750080	0.0084773		-0.0178795	0.0001595
	-0.0422618	0.0014519		-0.2514783	0.0010944
	-0.0602186	0.0024115		-0.0546466	0.0002984
	0.0643765	0.0021106		0.0275084	0.0001323
	-0.0327867	0.2107177		0.0048114	0.2261284

a higher number of equations—10 equations for the interval arithmetic benchmarks, 6 equations for the neuropsychology example, 5 equations for the chemical equilibrium application, 8 equations for the kinematic application, 10 equations for the combustion application, and 20 equations for the economics modeling.

Since we transformed a system of equations into a multiobjective optimization problem, whose number of objectives is equal to the number of equations contained by the corresponding system, our task is to deal with complicated high-dimensional optimization problems. The goal is to obtain values as close to zero as possible for each of the involved objectives. As evident from the obtained empirical results, the proposed approach is very much appealing for solving high-dimensional equations systems. As a measure of quality for the solutions obtained, the sum of the absolute values of the objectives (which are the modified equations of the initial system) is considered. The closer the value of this sum to zero, the better the solution.

From the graphical illustrations provided in the manuscript, it can be deduced that the proposed approach could obtain very good results even for some complicated systems such as combustion application, neuropsychology application, chemical equilibrium application, and economic modeling.

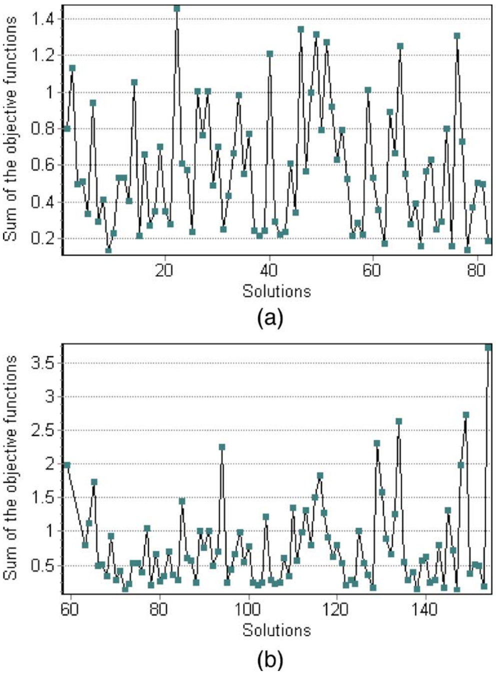

Fig. 11. Nondominated solutions obtained for the economics application. (a) Solutions whose sum of the absolute values of the objective functions is less than or equal to 1.5. (b) Sum of the absolute values of the objective functions for all the nondominated solutions obtained.

TABLE XX
CPU TIME REQUIRED BY THE EVOLUTIONARY ALGORITHM FOR ALL THE CONSIDERED BENCHMARKS

Benchmark	Running time (s)
Example 1	5.14
Example 2	5.09
Interval i1	39.07
Neurophysiology	28.90
Chemical equilibrium	32.71
Kinematics kin2	221.29
Combustion	151.12
Economics	640.92

The proposed method could be extended for more higher dimensional systems, although this will also involve increased computational complexity. In a similar manner, we can also solve inequality systems and systems of differential equations, which are part of our future research work.

REFERENCES

[1] A. Abraham, L. Jain, and R. Goldberg, Eds., *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. New York: Springer-Verlag, 2005.

[2] M. A. Abido, “Multiobjective evolutionary algorithms for electric power dispatch problem,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 315–329, Jun. 2006.

[3] R. Angira and B. V. Babu, “Multi-objective optimization using modified differential evolution (MDE),” *Int. J. Math. Sci.*, vol. 5, no. 2, pp. 371–387, 2006.

[4] B. V. Babu, P. G. Chakole, and J. H. Syed Mubeen, “Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor,” *Chem. Eng. Sci.*, vol. 60, no. 17, pp. 4822–4837, Sep. 2005.

[5] B. W. Bader, “Tensor–Krylov methods for solving large-scale systems of nonlinear equations,” *SIAM J. Numer. Anal.*, vol. 43, no. 3, pp. 1321–1347, 2005.

[6] A. Benedetti, M. Farina, and M. Gobbi, “Evolutionary multiobjective industrial design: The case of a racing car tire-suspension system,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 230–244, Jun. 2006.

[7] C. Brezinski, *Projection Methods for Systems of Equations*. Amsterdam, The Netherlands: Elsevier, 1997.

[8] C. G. Broyden, “A class of methods for solving nonlinear simultaneous equations,” *Math. Comput.*, vol. 19, no. 92, pp. 577–593, Oct. 1965.

[9] L. Chen, J. McPhee, and W. G. Yeh, “A diversified multiobjective GA for optimizing reservoir rule curves,” *Adv. Water Res.*, vol. 30, no. 5, pp. 1082–1093, May 2007.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*. Philadelphia, PA: SIAM, 2000.

[11] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.

[12] K. Deb, K. Mitra, R. Dewri, and S. Majumdar, “Towards a better understanding of the epoxy-polymerization process using multi-objective evolutionary computation,” *Chem. Eng. Sci.*, vol. 59, no. 20, pp. 4261–4277, Oct. 2004.

[13] J. E. Denis, “On Newton’s method and nonlinear simultaneous replacements,” *SIAM J. Numer. Anal.*, vol. 4, pp. 103–108, 1967.

[14] J. E. Denis, “On Newton like methods,” *Numer. Math.*, vol. 11, no. 4, pp. 324–330, May 1968.

[15] J. E. Denis, “On the convergence of Broyden’s method for nonlinear systems of equations,” *Math. Comput.*, vol. 25, no. 115, pp. 559–567, Jul. 1971.

[16] J. E. Denis and H. Wolkowicz, “Least change secant methods, sizing, and shifting,” *SIAM J. Numer. Anal.*, vol. 30, pp. 1291–1314, 1993.

[17] J. E. Denis, M. El Alem, and K. Williamson, “A trust-region algorithm for least-squares solutions of nonlinear systems of equalities and inequalities,” *SIAM J. Opt.*, vol. 9, no. 2, pp. 291–315, 1999.

[18] S. Effati and A. R. Nazemi, “A new method for solving a system of the nonlinear equations,” *Appl. Math. Comput.*, vol. 168, no. 2, pp. 877–894, Sep. 2005.

[19] L. J. Eshelman, R. A. Caruna, and J. D. Schaffer, “Biases in the crossover landscape,” in *Proc. Int. Conf. Genetic Algorithms*, J. Schaffer, Ed., 1989, pp. 10–19.

[20] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[21] T. N. Grapsa and M. N. Vrahatis, “The implicit function theorem for solving systems of nonlinear equations in R2,” *Int. J. Comput. Math.*, vol. 28, no. 1–4, pp. 171–181, 1989.

[22] T. N. Grapsa and M. N. Vrahatis, “A dimension-reducing method for solving systems of nonlinear equations in R^n,” *Int. J. Comput. Math.*, vol. 32, no. 3/4, pp. 205–216, 1990.

[23] T. N. Grapsa, M. N. Vrahatis, and T. C. Bountis, “Solving systems of nonlinear equations in R^n using a rotating hyperplane in R^{n+1},” *Int. J. Comput. Math.*, vol. 35, no. 1–4, pp. 133–151, 1990.

[24] T. N. Grapsa and M. N. Vrahatis, “A new dimension-reducing method for solving systems of nonlinear equations,” *Int. J. Comput. Math.*, vol. 55, no. 3/4, pp. 235–244, 1995.

[25] T. N. Grapsa and M. N. Vrahatis, “A dimension-reducing method for unconstrained optimization,” *J. Comput. Appl. Math.*, vol. 66, no. 1/2, pp. 239–253, Jan. 1996.

[26] T. N. Grapsa and M. N. Vrahatis, “Dimension reducing methods for systems of nonlinear equations and unconstrained optimization: A review,” *Recent Advances in Mechanics and Related Fields*, pp. 215–225, 2003.

[27] P. Van Hentenryck, D. McAllester, and D. Kapur, “Solving polynomial systems using a branch and prune approach,” *SIAM J. Numer. Anal.*, vol. 34, no. 2, pp. 797–827, Apr. 1997.

[28] M. J. Hirsch, C. N. Meneses, P. M. Pardalos, and M. G. C. Resende, “Global optimization by continuous grasp,” *Opt. Lett.*, vol. 1, no. 2, pp. 201–212, Mar. 2007.

[29] H. Hong and V. Stahl, “Safe starting regions by fixed points and tightening,” *Computing*, vol. 53, no. 3/4, pp. 323–335, Sep. 1994.

[30] J. M. Martinez, “Algorithms for solving nonlinear systems of equations,” in *Continuous Optimization: The State of the Art*, E. Spedicato, Ed. Norwell, MA: Kluwer, 1994, pp. 81–108.

[31] A. L. Medaglia, S. B. Graves, and J. L. Ringuest, “A multiobjective evolutionary approach for linearly constrained project selection under uncertainty,” *Eur. J. Oper. Res.*, vol. 179, no. 3, pp. 869–894, Jun. 2007.

[32] K. Meintjes and A. P. Morgan, “Chemical equilibrium systems as numerical test problems,” *ACM Trans. Math. Softw.*, vol. 16, no. 2, pp. 143–151, Jun. 1990.

[33] R. E. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA: SIAM, 1979.

[34] A. P. Morgan, "Computing all solutions to polynomial systems using homotopy continuation," *Appl. Math. Comput.*, vol. 24, no. 2, pp. 115–138, Nov. 1987.

[35] A. P. Morgan, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[36] P. Y. Nie, "A null space method for solving system of equations," *Appl. Math. Comput.*, vol. 149, no. 1, pp. 215–226, Feb. 2004.

[37] P. Y. Nie, "An SQP approach with line search for a system of nonlinear equations," *Math. Comput. Model.*, vol. 43, no. 3/4, pp. 368–373, Feb. 2006.

[38] J. Nielson and B. Roth, "On the kinematic analysis of robotic mechanisms," *Int. J. Robot. Res.*, vol. 18, no. 12, pp. 1147–1160, Dec. 1999.

[39] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1970.

[40] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[41] G. Quan, G. W. Greenwood, D. Liu, and S. Hu, "Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms," *Eur. J. Oper. Res.*, vol. 177, no. 3, pp. 1969–1984, Mar. 2007.

[42] P. Reed, J. B. Kollat, and V. K. Devireddy, "Using interactive archives in evolutionary multiobjective optimization: A case study for long-term groundwater monitoring design," *Environ. Model. Softw.*, vol. 22, no. 5, pp. 683–692, May 2007.

[43] W. C. Rheinboldt, *Methods for Solving Systems of Equations*. Philadelphia, PA: SIAM, 1974.

[44] B. Salimbahrami and B. Lohmann, "Order reduction of large scale second-order systems using Krylov subspace methods," *Lin. Alg. Appl.*, vol. 415, no. 2/3, pp. 385–405, Jun. 2006.

[45] J. D. Schaffer and A. Morishima, "An adaptive crossover distribution mechanism for genetic algorithms," in *Proc. Int. Conf. Genetic Algorithms*, Hillsdale, NJ, 1987, pp. 36–40.

[46] W. M. Spears and K. A. De Jong, "On the virtues of uniform crossover," in *Proc. Int. Conf. Genetic Algorithms*, 1991, pp. 230–236.

[47] G. Syswerda, "Uniform crossover in genetic algorithms," in *Proc. Conf. Genetic Algorithms*, Los Altos, CA, 1989, pp. 2–9.

[48] R. E. Steuer, *Multiple Criteria Optimization. Theory, Computation, and Application*, ser. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. New York: Wiley, 1986.

[49] K. C. Tan, Y. J. Yang, and C. K. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 527–549, Oct. 2006.

[50] J. Verschelde, P. Verlinden, and R. Cools, "Homotopies exploiting Newton polytopes for solving sparse polynomial systems," *SIAM J. Numer. Anal.*, vol. 31, no. 3, pp. 915–930, Jun. 1994.

Crina Grosan received the M.S. degree in mathematics and the Ph.D. degree in computer science from Babes-Bolyai University, Cluj Napoca, Romania, in 2005.

She is currently a Lecturer of artificial intelligence with Babes-Bolyai University and was a Researcher with the Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence, Norwegian University of Science and Technology, Trondheim, Norway. She has published over 100 research papers in peer-reviewed international journals, book chapters, and conference proceedings. She is the coauthor of two books on programming languages and is also a coeditor of four volumes titled *Stigmergic Optimization*, *Swarm Intelligence in Data Mining*, *Hybrid Evolutionary Algorithms*, and *Engineering Evolutionary Intelligent Systems*. She has guest-edited a special issue on soft computing for modeling and simulation for the *International Journal of Simulation Systems, Science & Technology* (U.K. Simulation Society). Her research focuses on different global optimization techniques and applications. She has been researching such topics as multiobjective optimization, global optimization, operations research, numerical analysis, computational intelligence, and swarm intelligence. She is the Managing Editor of the *International Journal of Computational Intelligence Research* and also serves as a member of the editorial board of a few other journals.

Dr. Grosan cofounded the Evolutionary Multiobjective Optimization: Design and Applications International Workshop Series in 2005.

Ajith Abraham (M'96–SM'07) received the Ph.D. degree from Monash University, Melbourne, Australia, in 2001.

He is currently a Visiting Professor with the Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Centre of Excellence, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. Before joining NTNU, he was working under the South Korean Government's Institute of Information Technology Advancement (IITA) Professorship Program at Yonsei University, Seoul, Korea, and Chung-Ang University, Seoul. He was a Visiting Researcher with Rovira i Virgili University, Tarragona, Spain, during 2005–2006 and is currently an Adjunct Professor with Jinan University, Jinan, China, and Dalian Maritime University, Dalian, China. He has authored or coauthored more than 350 research publications in peer-reviewed reputed journals, book chapters, and conference proceedings. His primary research interests are in computational intelligence, with a focus on using global optimization techniques for designing intelligent systems. His application areas include Web services, information security, Web intelligence, financial modeling, multicriteria decision making, data mining, etc. He is a regular reviewer of IEEE Intelligent Systems, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, and IEEE TRANSACTIONS ON POWER SYSTEMS.

Dr. Abraham serves on the Editorial Boards of more than 30 international journals and has also guest edited 26 special issues on various topics for international journals. Since 2001, he has been actively involved in the Hybrid Intelligent Systems and the Intelligent Systems Design and Applications series of annual international conferences. He was the recipient of five Best Paper Awards.