Solving Nonlinear Equation Systems Using Evolutionary Algorithms

Crina Grosan
Department of Computer Science
Babeş-Bolyai University
Cluj-Napoca, 3400, Romania
cgrosan@cs.ubbcluj.ro

Ajith Abraham
School of Computer Science and Engineering
Chung-Ang University
Seoul, 156-756 Korea
ajith.abraham@ieee.org

ABSTRACT

This paper proposes a new perspective for solving systems of nonlinear equations. A system of equations can be viewed as a multiobjective optimization problem: every equation represents an objective function whose goal is to minimize difference between the right and left term of the corresponding equation in the system. We used an evolutionary computation technique to solve the problem obtained by transforming the system of nonlinear equations into a multiobjective problem. Results obtained are compared with a very new technique [10] and also some standard techniques used for solving nonlinear equation systems. Empirical results illustrate that the proposed method is efficient.

1. INTRODUCTION

A nonlinear system of equations is defined as:

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

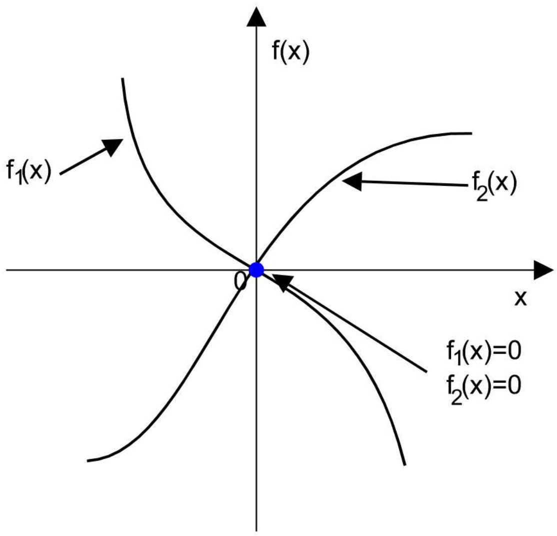
$x = (x_1, x_2, \ldots, x_n)$, refers to n equations and n variables; where $f_1, \ldots, f_n$ are nonlinear functions in the space of all real valued continuous functions on $\Omega = \prod_{i=1}^{n} [a_i, b_i] \subset \Re^n$.

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:

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

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

Figure 1: Example of solution in the case of a two nonlinear equation system represented by f_1 and f_2.



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

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

is having 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 looking for are those points (if any) that are common to the zero contours of $f_i, i = 1, ..., n$.

There are several ways to solve nonlinear equation systems ([1], [5]-[9], [13]). Probably, the most popular techniques are the Newton type techniques. Other techniques are:

- Trust-region method [3];
- Broyden method [2];
- Secant method [12];
- Halley method [4].

GECCO '06 Seattle, WA, USA

Newton's method

We can approximate f using the first order Taylor expansion in a neighborhood of a point $x^k \in \Re^n$. The Jacobian matrix $J(x^k) \subset \Re^{nxn}$ to $f(x)$ evaluated at x^k is given by:

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

Then:

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

By setting the right side of the equation to zero and neglecting terms of higher order (except the first) ($O(||p||^2)$) we obtain:

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

Then, the Newton algorithm is described as follows:

: Algorithm1: Newton algorithm

Set k=0

Guess an approximate solution x^0.

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 converged 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 which is reasonably close to the true zero, then replace the function by its tangent and computes 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 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).
4. The Jacobian matrix is needed but in many problems analytical derivatives are unavailable.
5. If function evaluation is expensive, then the cost of finite-difference determination of the Jacobian can be prohibitive.

Effati's method

In [10], Effati and Nazemi proposed a new method for solving systems of nonlinear equations. The method proposed in [10] is presented below.

The following notations are used:

$x_i(k+1) = f_i(x_1(k), x_2(k), ..., x_n(k))$;

$f(x_k) = (f_1(x_k), f_2(x_k), ..., f_n(x_k))$;

$i = 1, 2..., n$ and $x_i : N \to \Re$.

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

Define:

$u(k) = (u_1(k), u_2(k), ..., u_n(k))$.

$x(k+1) = u(k)$

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

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

The error function E is defined as follows:

$E[x^t, u^t] = \sum_{k=0}^{t-1} f^0(x(k), u(k))$.

$x^t = (x(1), x(2), ..., x(t-1), 0)$

$u^t(u(1), u(2), ..., u(t-1), 0)$.

Consider the following problem:

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

In the theorem illustrated by Effati and Nazemi [10] 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 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 first constructed. From there, x^t could be obtained (see for details [10]). The measure theory method is improved in [10]. The interval $[1, t]$ is divided into the subintervals $S_1 = [1, t-1]$ and $S_2 = [t-1, t]$. The problem P_1 is solved in both subintervals and two errors E_1 and E_2 respectively are obtained. 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.

2. PROBLEM TRANSFORMATION

This section explains how the problem is transformed to a multiobjective optimization problem. First, the basic definitions of a multiobjective optimization problem is presented and what it denotes an optimal solution for this problem [15].

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

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

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

The multiobjective optimization problem is defined as:

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

For deciding wether 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, y be two decision vectors (solutions) from Ω.

Solution x *dominate* y (also written as $x \succ y$) if and only if the following conditions are fulfilled:

(i) $f_i(x) \geq f_i(y), \forall i = 1, 2, \ldots, n$,

(ii) $\exists j \in \{1, 2, \ldots, n\}$: $f_j(x) > f_j(y)$.

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 Pareto optimal or minimal solutions, including words such as nondominated, noninferior, efficient, 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 *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 left side and right side of the equation. Since the right term is zero, the objective function is denoted by the absolute value of the left term.

The system (P) is then equivalent to:

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

3. EVOLUTIONARY NONLINEAR EQUATION SYSTEM

An evolutionary technique is applied to solving the multiobjective problem obtained by transforming the system of equations. We generate some starting points (initial solutions) within defined domain. Then these solutions were evolved in an iterative manner. In order to compare two solutions we use the Pareto dominance relationship. Genetic operators (such as crossover and mutation) are used. Convex crossover and gaussian mutation are used [11]. An external set was used for storing all the nondominated solutions found during the iteration process. Tournament selection is applied. n individuals are randomly selected from the unified set of current population and external population. Out of these n solutions the one which dominated a greater number of solutions is selected. If there are two or more 'equal' solutions then one is picked at random. At each iteration we update this set by introducing all the nondominated solutions obtained at the respective step and we are removing form the external set all solutions which will become dominated.

The algorithm can be described as follows:

Step 1. Set $t = 0$.
Randomly generate population $P(t)$.
Set $EP(t) = \emptyset$. (EP denoted the external population.
Step 2. **Repeat**
Step 2.1. Evaluate $P(t)$.
Step 2.2. Selection $(P(t) \cup (t))$.
Step 2.3. Crossover.
Step 2.4. Mutation.
Step 2.3. Select all nondominated individuals obtained.
Step 2.3. Update $EP(t)$.
Step 2.3. Update $(P(t)$ (keep best between parents and offspring).
Step 2.3. Set $t := t + 1$.
Until $t = number_of_generations$.
Step 3. Print $EP(t)$.

4. EXPERIMENTS

This section reports the several experiments and comparisons which we performed. We consider the same problems (Example 1 and Example 2 below) as the ones used by Effati and Nazemi [10]. Parameters used by the evolutionary approach for both examples are given in Table 1.

Table 1: Parameter setting used by the evolutionary approach.

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

4.1 Example 1

Consider the following nonlinear system:

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

Results obtained by applying Newton's method, Effati's technique and the proposed method are presented in Table 2.

As evident from Table 2, results obtained by the Evolutionary Algorithm (EA) are better than the ones obtained by the other techniques. Also, by applying an evolutionary technique we don't need any additional information about the problem (such as the functions to be differentiable, an adequate selection of the starting point, etc).

Table 2: Results for the first example.

Method	Solution	Functions values
Newton	(0.15, 0.49)	(-0.00168, 0.01497)
Effati	(0.1575, 0.4970)	(0.005455, 0.00739)
EA	(0.15772, 0.49458)	(0.001264, 0.000969)

4.2 Example 2

We have the following problem:

$$\begin{cases} f_1(x_1, x_2) = e^{x_1} + x_1x_2 - 1 = 0 \\ f_2(x_1, x_2) = sin(x_1x_2) + x_1 + x_2 - 1 = 0 \end{cases}$$

Results obtained by Effati's method and the evolutionary approach are given in Table 3. For this example, the evolutionary approach obtained better results better than Effati's method. These experiments show the efficiency and advantage of applying evolutionary techniques for solving systems of nonlinear equations against standard mathematical approaches.

Table 3: Results for the second example.

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

5. CONCLUSIONS

The proposed approach seems to be very efficient for solving equation systems. We analyzed here the case of nonlinear equation systems. The proposed approach could be extended for higher dimensional systems. Also, in a similar manner, we can solve inequations systems.

6. REFERENCES

[1] C. Brezinski, Projection methods for systems of equations, Elsevier, 1997.

[2] C.G. Broyden, A class of methods for solving nonlinear simultaneous equations. Mathematics of Computation, 19, 577-593, 1965.

[3] A.R. Conn, N.I.M. Gould, P.L. Toint, Trust-Region methods, SIAM, Philadelphia, 2000.

[4] A. Cuyt, P.van der Cruyssen, Abstract Pade approximants for the solution of a system of nonlinear equations, Computational Mathematics and Applications, 9, 139-149, 1983.

[5] J.E. Denis, On Newtons Method and Nonlinear Simultaneous Replacements, SIAM Journal of Numerical Analisys, 4, 103108, 1967.

[6] J.E. Denis, On Newtonlike Methods, Numerical Mathematics, 11 , 324330, 1968.

[7] J.E. Denis, On the Convergence of Broydens Method for Nonlinear Systems of Equations, Mathematics of Computation, 25, 559567, 1971.

[8] J.E. Denis, H. Wolkowicz, LeastChange Secant Methods, Sizing, and Shifting, SIAM Journal of Numerical Analisys, 30, 12911314, 1993.

[9] J.E. Denis, M. ElAlem, K. Williamson, A Trust-Region Algorithm for Least-Squares Solutions of Nonlinear Systems of Equalities and Inequalities, SIAM Journal on Optimization 9(2), 291-315, 1999.

[10] S. Effati, A.R. Nazemi, A new methid for solving a system of the nonlinear equations, Applied Mathematics and Computation, 168, 877-894, 2005

[11] Goldberg, D.E. Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading, MA, 1989.

[12] W. Gragg, G. Stewart, A stable variant of the secant method for solving nonlinear equations, SIAM Journal of Numerical Analisys, 13, 889-903, 1976.

[13] J. M. Ortega and W. C. Rheinboldt, Iterative solution of nonlinear equations in several variables. New York: Academic Press, 1970

[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 2002.

[15] Steuer, R. E. Multiple Criteria Optimization. Theory, Computation, and Application. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. New York: John Wiley Sons, Inc, 1986.