PERGAMON

Nonlinear Analysis 47 (2001) 3437–3442

Nonlinear Analysis

www.elsevier.nl/locate/na

Evolutionary algorithms for computing zeros of nonlinear functions

G.A. Tsirogiannis [a,b,d], G.N. Beligiannis [a,c,d], S.D. Likothanassis [a,c,d], M.N. Vrahatis [b,d]

[a] *Department of Computer Engineering & Informatics, University of Patras, GR–26500 Patras, Greece*

[b] *Department of Mathematics, University of Patras, GR–26110 Patras, Greece*

[c] *Computer Technology Institute (C.T.I.), Kolokotroni 3, GR–26221 Patras, Greece*

[d] *University of Patras Artificial Intelligence Research Center (UPAIRC)*

Abstract

A method for locating and computing solutions of systems of nonlinear algebraic and/or transcendental equations or fixed points of continuous functions is described. Our method is based on various well–known notions of Combinatorial Topology and it utilizes evolutionary programming techniques. In particular, the proposed method constructs a Sperner simplex in the n–dimensional Euclidean space by applying an evolutionary programming technique. Our method converges rapidly to a solution, independently of the initial guess, and is particularly useful, since it proceeds solely by comparing relative sizes of the function values.

1 Introduction

Many problems require the solution of the equation $F_n(x) = \Theta^n$, where $\Theta^n = (0,\ldots,0)$ is the origin of $\mathbb{R}^n$ and $F_n = (f_1, f_2, \ldots, f_n) : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}^n$, is a continuous nonlinear function from a domain $\mathcal{D} \subset \mathbb{R}^n$ into $\mathbb{R}^n$. Obviously the problem of solving the above system is similar to the problem of computing fixed points, i.e. the solution of the system $G_n(x) = F_n(x) - x = \Theta^n$, gives a point $x^* \in \mathcal{D}$ such that $F_n(x^*) = x^*$ which, of course is a fixed point of F_n. These systems of nonlinear equations arise in a large number of applications in many scientific

PII: S0362-546X(01)00460-6

and technological fields, including mathematics, physical sciences, medicine, statistics, operation research, business administration, economics, system analysis and computer science, for which a solution (or sometimes all solutions) is of practical significance.

Methods mainly of contraction mapping type, such as Newton's method and related classes of algorithms [9], require the starting point to be within the immediate vicinity of the eventual solution. The necessity of having a good approximation to the value of an unknown solution is obviously a severe disadvantage. Furthermore, in many cases, these methods fail, due to the nonexistence of derivatives or poorly behaved partial derivatives. Also, Newton's method, as well as Newton-like methods, often converge to a solution almost independently of the initial guess, while there may exist several solutions nearby, all of which are desired for the application. For the fractal like geometry of the basin of convergence of these methods see [2,3].

In this contribution, a new evolutionary approach is presented which combines the effectiveness and efficiency of the Evolutionary Computation techniques [11] with some concepts of the Combinatorial Topology, aiming to deal with the problem of locating and computing the fixed points of a nonlinear function. Our method proceeds solely by comparing the relative sizes of the function values. The proposed method has been implemented and tested and the corresponding numerical results indicate that our method is an efficient and effective one.

2 Fixed points theorems and labeling lemmas

One of the most important theorems in the field of nonlinear equations is Brouwer's fixed point theorem. If we rewrite a system of nonlinear equations in fixed point form, then the theorem states that, under mild assumptions, we will have a fixed point, i.e. a solution. This theorem has been used for many years to prove the existence of a solution of complicated systems of nonlinear equations [5,10,15–17].

Brouwer's fixed point theorem [4] states that: any continuous mapping $F_n : \sigma^n \to \sigma^n$ from an n-simplex $\sigma^n \subset \mathbb{R}^n$ into itself has at least one fixed point x^*, that is $F_n(x^*) = x^*$. A proof of Brouwer's theorem for the simplex was given by Knaster, Kuratowski and Mazurkiewicz [7]. The Knaster, Kuratowski and Mazurkiewicz covering lemma states that: if $C_i, i \in \mathbb{N}_0 = \{0, 1, \dots, n\}$ is a family of closed subsets of σ^n satisfying the following conditions:

(1) $\sigma^n = \bigcup_{i\in\mathbb{N}_0} C_i$ and
(2) if $0 \neq I \subseteq \mathbb{N}_0$ and $J = \mathbb{N}_0 - I$ then $\bigcap_{i\in I} \sigma^{n,i} \subseteq \bigcup_{j\in J} C_j$,

then it holds that $\bigcap_{i\in\mathbb{N}_0} C_i \neq 0$, where $\sigma^{n,i} = \{v^0, v^1, \dots, v^{i-1}, v^{i+1}, \dots, v^n\}$ determines the ith face of σ^n.

Scarf and Hansen proved a lemma similar to the above lemma. In addition, an interesting generalization of the Knaster, Kuratowski and Mazurkiewicz lemma has been given by Gale [6].

The Sperner lemma is the basis for a proof of the Brouwer fixed point theorem [12,13]. Before stating the Sperner lemma we give several concepts that are needed:

Let $v^0, v^1, \ldots, v^n$ denote the vertices of σ^n. A k–face of σ^n determined by the vertices $v^{i0}, v^{i1}, \ldots,\ v^{ik}$ is called the carrier of a point v if v lies on this k–face and not on any subface of this k–face. A function $\lambda(v)$ defined on a σ^n is called a proper labeling function if it satisfies the following conditions [1]:

a) $\lambda(v) \in \{0, 1, ..., n\}$,
b) $\{\lambda(v^0), \lambda(v^1), \ldots, \lambda(v^n)\} = \{0, 1, \ldots, n\}$,
c) If the i–face, determined by the vertices $v^{i0}, v^{i1}, \ldots, v^{ik}$, is the carrier of v then $\lambda(v) \in \{\lambda(v^{k0}), \lambda(v^{k1}), \ldots, \lambda(v^{ki})\}$.

Let $\{\omega^0, \omega^1, \ldots, \omega^k\}$ denote the vertices of a k–simplex, $k \leq n$, of a simplicial subdivision of σ^n. This k–simplex is said to have a complete set of labels if $\{\lambda(\omega^0), \lambda(\omega^1), \ldots,\ \lambda(\omega^k)\} = \{0, 1, ..., k\}$ holds.

The Sperner's lemma [12] states that: for any simplicial subdivision and proper labeling function of σ^n there is at least one n–simplex of the subdivision with a complete set of labels.

A Sperner simplex is this n–simplex with a complete set of labels. For well–behaved continuous functions and a fine enough simplicial subdivision, the vertices of such a simplex approximate a fixed point or a root of the mapping. One can use Sperner's lemma to give a constructive proof of Brouwer's fixed point theorem. Yoseloff [19] proved that the Sperner's lemma can be derived from Brouwer's fixed point theorem, and therefore they are equivalent.

Sperner [14] gave a very general labeling lemma that states the following: let the n–simplex σ^n be triangulated. Label each vertex of the simplices in the triangulation by an integer from the set $\{0, 1, ..., n\}$. Then the number of $(n-1)$-simplices on the boundary with labels $\{0, 1, ..., n-1\}$ is equal to the number of n–simplices in the interior with labels $\{0, 1, ..., n\}$. All simplices are counted with orientation.

In this article we use the labeling function $\lambda(x) : \mathbb{R}^n \to \{0, 1, \ldots, n\}$ with values $\lambda(x) = \min\{i : f_i(x) \leq x_i\}$ if the inequality holds for some i, or $\lambda(x) = 0$ otherwise.

3 The proposed evolutionary approach and experimental results

In this section we present our evolutionary approach that can be applied to any dimension. Our aim is to find a Sperner simplex by using the labeling function λ described in the previous section. Next we present a high level description of the proposed evolutionary approach:

1) We use $(n+1)$ Evolutionary Algorithms in order to find points in $\mathbb{R}^n$ which are labeled with all the elements of the set $\{0, 1, ..., n\}$. We have, of course, bounds

for every $x_i \in \mathbb{R}$.

2) From the previous step we find n_i points which can be labeled with the label $i \in \mathbb{N}_0$ where $\mathbb{N}_0 = \{0, 1, ..., n\}$. We select a point that can be labeled with the label i, a point that can be labeled with the label $j, \dots,$ and finally a point that can be labeled with the label k and we check if this face satisfies the Sperner criterion for labeling vertices. The reason for doing so is that a Sperner simplex is a hint that this simplex may obey the Knaster, Kuratowski, Mazurkiewicz covering lemma. In this case as we have already seen, this determines that the simplex contains with certainty at least one fixed point.
3) If there is a simplex S^m which is a Sperner simplex then we apply the former methodology for the search space defined by this simplex.

Details of the algorithm:

(1) *First step of the algorithm*: It is the step where the $(n+1)$ Evolutionary Algorithms are used and the proper fitness function is based on the function λ. As we observe, the function λ consists of a set of inequalities. The value of λ ($\lambda \in \mathbb{N}_0$) depends on the number of the inequalities that are satisfied. So, we can use the sum of absolute values of the differences $f_i(x) - x_i$ as the fitness function of our Evolutionary Algorithm. In this way, the inequalities that are satisfied are not considered at all, while the ones that are not satisfied tend to do so. This happens because, as the value of the difference $|f(x_i) - x_i|$ decreases, the value of the fitness function decreases, too. The lower bound of the fitness function will be zero. So, all the members of the population having zero value for their fitness function are suitable to be labeled with the label i.

(2) *Second step of the algorithm*: We choose $(n+1)$ points labeled with different colors in order to form a Sperner simplex.

(3) *Third step of the algorithm*: If we find $(n+1)$ points that form a Sperner simplex then we can either find the n–dimensional box which contains the simplex we have found and execute the method iteratively with the new bounds, or specify the search space defined by the simplex and execute Step 1 for this space using again the new constrains that have been calculated. When the size of the simplex is small enough we invoke the classical EA to achieve a better approximation. Notice that a Sperner simplex does not guarantee that a fixed point of the function is contained within it. It is only a hint of it. The reason of using the Sperner simplex is to shrink the search space. When we reach to an area small enough, the following EA can be applied: Its fitness function consists of the summation of the squares of the functional value of the ith equation minus the value of the ith variable.

The approach we use from the Combinatorial Topology reformulates the search space in a simpler form which is more easily studied. Furthermore, the Evolutionary Computation techniques we apply take advantage every time of all the characteristics of the search space and result in quick convergence.

The proposed evolutionary algorithm has been implemented and tested and its performance has been compared with classical EAs. Our experience is that the

Table 1
The simulation results obtained from the classical EA.

Function	Success	μ	σ	min	max
Complex	40/100	6118	1761.282	2020	7204
Freudenstein	10/100	6981.3	270.712	6680	7826
Himellblau	66/100	4540	1749.491	2056	7390
Werner–Weber	0/100				

algorithm behaves predictably and reliably and the results are quite satisfactory. All the EAs used have been implemented using the Genetic Algorithms' library GEATbx and all the parameters used have been set to their default values.

In our approach the fitness function for the corresponding EAs used has the following form: It selects the points of the search space which satisfy (or nearly satisfy) the conditions of λ in such a way that the point is labeled with the label i. If an inequality is satisfied we do not take it into consideration at all. If all the inequalities are satisfied then the fitness function of the respective point becomes zero. For instance, for the label 2 the fitness function (ff) takes the values:

if $f_1(x) > x_1$ then $ff = 0$ else $ff = |f_1(x) - x_1|$,
if $f_2(x) \leq x_2$ then $ff = 0$ else $ff = ff + |f_2(x) - x_2|$.

In this case, we obtain the three sets: S_0={points with label 0}, S_1={points with label 1}, S_2={points with label 2}. If we choose a point from every set then we are likely to obtain Sperner simplices. When a simplex is found then we search within the region defined by it. The constraints of this region can very easily be included in the fitness function as in the case of the function λ. We have implemented this method using an EA with genes forming a simplex. The aim of this EA is to find a Sperner simplex. The fitness function is the same as the one described previously extended by the summation of all the objective values of the EA used to determine the colors 0, 1 and 2.

Next, we give quantitative results obtained by applying classical EAs and our method to four test functions [2,8,18] in the region $[-1000, 1000] \times [-1000, 1000]$. All experiments have been conducted 100 times and the corresponding results are exhibited in Tables 1 and 2. The results are given in terms of the number of successful runs out of 100 (Success), the average number of function evaluations (μ) required to obtain convergence, the corresponding standard deviation (σ) and

Table 2
The simulation results obtained from our hybrid method.

Function	Success	μ	σ	min	max
Complex	57/100	3757	779.882	2802	5106
Freudenstein	30/100	5831	1360.047	4428	8932
Himellblau	80/100	4852.25	1739.463	2730	7992
Werner–Weber	60/100	6091	2011.99	4124	9516

the minimum (min) and maximum (max) number of function evaluations.

4 Conclusions

In this work an evolutionary approach is presented, which combines the effectiveness of the well-known and widely used Evolutionary Algorithms with theoretical results for the existence of a fixed point obtained by means of the Combinatorial Topology. Our approach is able to provide solution to very difficult problems that appear in many scientific and technological areas and real world applications since it proceeds solely by comparing relative sizes of the function values. The method can be implemented in parallel, since the EAs are naturally parallel structured, thus increasing the computational speed.

References

[1] E.L. Allgower and C.L. Keller, Computing 8 (1971) 157.

[2] G.S. Androulakis and M.N. Vrahatis, J. Comput. Appl. Math. 72 (1996) 41.

[3] G.S. Androulakis, G.D. Magoulas and M.N. Vrahatis, Nonlinear Anal. T.M.A. 30 (1997) 4539. Institute of Physics Publishing, 1997.

[4] L.E.J. Brouwer, Mathematisce Annalen 71 (1912) 97.

[5] W. Foster, Homotopy methods: Handbook of Global Optimization, R. Horst and P.M. Pardalos (eds.), Kluwer Academic Publishers, 1995.

[6] D. Gale, Inter. J. Game Theory 13 (1984) 61.

[7] B. Knaster, K. Kuratowski and S. Mazurkiewicz, Fundamenta Math. 14 (1929) 132.

[8] J.J. Moré, B.S. Garbow and K.E. Hillstrom, ACM Trans. Math. Softw. 7 (1981) 17.

[9] J.M. Ortega and W.C. Rheinbolt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, 1970.

[10] H. Scarf, SIAM J. Appl.Math 15 (1967) 1328.

[11] H.P Schwefel, Evolution and Optimum Seeking, J. Willey & Sons, 1995.

[12] E. Sperner, Abh. Math. Sem. Univ. Hamburg 6 (1928) 265.

[13] E. Sperner, Abh. Math. Sem. Univ. Hamburg 10 (1934) 1.

[14] E. Sperner, Numerical solution of highly nonlinear problems, 1980.

[15] M.G. Todd, The computation of fixed points and applications, Lecture notes in Economics and Mathematical Systems, Mathematical Economics, M. Beckmann, H.P. Kunzi (eds.), Springer–Verlag, 1976.

[16] M.N. Vrahatis, Proc. Amer. Math. Soc. 107 (1989) 701.

[17] M.N. Vrahatis, Simplex Bisection and Sperner Simplices (2000), to appear.

[18] M.N. Vrahatis, G.S. Androulakis and G.E. Manoussakis J. Math. Anal. Appl., 197 (1996) 586.

[19] M. Yoseloff, J. Comb. Theory (A) 17 (1974) 95.