Nonlinear Analysis: Real World Applications 11 (2010) 2465–2471

Contents lists available at ScienceDirect

Nonlinear Analysis: Real World Applications

journal homepage: www.elsevier.com/locate/nonrwa

On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods

I.G. Tsoulos [a,*], Athanassios Stavrakoudis [b]

[a] Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece

[b] Department of Economics, University of Ioannina, 45110 Ioannina, Greece

ARTICLE INFO

Article history:
Received 14 April 2008
Accepted 4 August 2009

Keywords:
Nonlinear equations
Global optimization
Stopping rules
Clustering

ABSTRACT

A novel method of locating all real roots of systems of nonlinear equations is presented here. The root finding problem is transformed to optimization problem, enabling the application of global optimization methods. Among many methods that exist in global optimization literature, Multistart and Minfinder are applied here because of their ability to locate not only the global minimum but also all local minima of the objective function. This procedure enables to locate all the possible roots of the system. Various test cases have been examined in order to validate the proposed procedure. This methodology does not make use of *a priori* knowledge of the number of the existing roots in the same manner as the corresponding global optimization methodology which does not make use of *a priori* knowledge of the existed number of local minima. Application of the new methodology resulted in finding all the roots in all test cases. The proposed methodology is general enough to be applied in any root finding problem.

1. Introduction

Many applied problems are reduced to solving systems of nonlinear equations, which is one of the most basic problems in mathematics. This task has applications in scientific fields such as physics [1–3], chemistry [4], economics [5] etc. There are several methods proposed in the literature to tackle this problem, however a complete solution has not yet been achieved. Recent paradigms include cases such as subdivision methods [6,8], exclusion test methods [9] Newton method [10–13,7], Trust region methods [14,15], Tensor methods [16,17], methods that utilize evolutionary algorithms [18,19] etc. Recently, Hirsch et al. have published another work [21] which estimates all the roots of systems of nonlinear equations by gradually adapting the minimization problem as roots are found.

A system of nonlinear equations may be defined as follows:

$$f(x)=\begin{pmatrix} f_1(x)\\ f_2(x)\\ \vdots\\ f_n(x)\end{pmatrix} \tag{1}$$

with $x\in S=\left[a_{1,}b_1\right]\times[a_2,b_2]\dots[a_n,b_n]\subset R^n$ and $f_1,f_2,\dots,f_n$ being nonlinear continuous functions, such that $f:S\to R$. Some of the equations can be linear but at least one of them cannot. A point $x\in S$ is called root of the system if every

* Corresponding author. Tel.: +30 2651098871.
E-mail address: itsoulos@gmail.com (I.G. Tsoulos).

1468-1218/$ – see front matter
doi:10.1016/j.nonrwa.2009.08.003

equation of the system is zero:

$$\begin{aligned} f_1(x) &= 0 \\ f_2(x) &= 0 \\ &\vdots \\ f_n(x) &= 0. \end{aligned} \tag{2}$$

This paper focuses on locating all the roots of systems of nonlinear equations using global optimization methods such as Multistart and Minfinder [20]. The original system of equations is transformed to an optimization problem and subsequently the global optimization methods attempt to estimate all the local minima of the optimization problem. This procedure enables the finding of all possible roots of the system.

Using optimization methods in order to solve a system of nonlinear equations has been also used successfully in the past. However, most of these approaches used local and deterministic optimization methodology. Taking into account the recent advantages in the area of stochastic and global optimization, we propose now a direct application of these methodologies in order to suggest a more general solution of the problem. A novel advantage of our proposed method is that it enables the application to any test case, since it does not make use of *a priori* knowledge of the number of roots in the system. Moreover, the incorporation of newly proposed fast global optimization methods [20] enables the location of all real roots of a system in extremely short CPU times. Also, the combination of global optimization methods with local optimization techniques increases the effectiveness of our approach.

The rest of this article is organized as follows: in Section 2 the methods Multistart and Minfinder are briefly discussed, in Section 3 the test problems as well as the results from the application of the methods to them are presented and finally in Section 4 some conclusions are derived and a discussion about future work is made.

2. Method description

2.1. Problem formulation

In order to use global optimization methods, the system of Eq. (2) is transformed to an optimization problem. This is achieved by using the auxiliary function:

$$F(x) = \sum_{i=1}^{n} f_i^2(x). \tag{3}$$

Squares are preferred from absolute values, because of their ability to provide analytical derivatives. By definition, $F(x) \geq 0$. Thus, for the global minimum x^* of $F(x)$ it holds $F(x^*) \geq 0$. If $\exists x^* : F(x^*) = 0$, then it implies that x^* is a global minimum and subsequently $f_1(x^*) = f_2(x^*) = \cdots = f_n(x^*) = 0$ and thus x^* is a root for the corresponding system of equations. Finding all the x^* such that $F(x^*) = 0$ corresponds to locating all the roots of the system. Of course, some of the local minima of $F(x)$ could have function value greater than zero. Such solutions are discarded from the algorithm, since they do not correspond to roots of the system.

The most used global optimization methods to locate all the local minima of $F(x)$ is the Multistart and the Minfinder methods described below.

2.2. Multistart

The Multistart method is the simplest global optimization method and it is the base for many others more efficient methods such as clustering methods. The main steps of any Multistart-like method are presented in Algorithm 1. Even though the Multistart method is quite simple is also quite ineffective when bad stopping rules are used and so a good stopping rule should be used that is effective and economical, i.e. locating all the local minima of the function using the least number of function evaluations. Lagaris and Tsoulos have proposed [22] three stopping rules for the Multistart method that are based on asymptotic considerations. The first stopping rule is called **Double-Box** and it uses a Monte Carlo based model that enables the determination of the coverage of the bounded search domain. The second method is called **Observables** stopping rule and it is based on a comparison between the expectation values of observables quantities to the actually measured ones. The third rule is called Expected Minimizers and is based on estimating the expected number of local minima in the specified domain. These stopping rules were used in our experiments for the location of the roots.

2.3. Minfinder

The second global optimization method used was the Minfinder method. This method is a new clustering algorithm that aims to locate all the local minima of a multidimensional continuous and differentiable function inside a bounded domain. The method utilizes the Double-Box stopping rule mentioned before and its main steps are presented in Algorithm 2. In this

Algorithm 1 The main steps of the Multistart method

1. **Set** $X^* = \emptyset$
2. **Set** the number of samples N.
3. **Set** iter $= 0$
4. **For** $i = 1..N$
   (a) Sample a point x in the feasible region of the objective function.
   (b) **Apply** a deterministic local procedure $L(x)$ yielding a local minimum x^*.
   (c) **If** $x^* \notin X^*$ **then** $X^* = X^* \cup x^*$.
5. **End For**
6. **Set** iter $=$ iter $+ 1$
7. **If** the termination criteria hold, **then** terminate **else goto** step 3.

Algorithm 2 The main steps of the Minfinder global optimization procedure

1. **Initialization step**:
   (a) Set the number of samples N
   (b) Set $X^* = \emptyset$ (local minimizers).
2. **Sampling step**:
   (a) $S = \emptyset$
   (b) **For** $i = 1..N$
      i. **Sample** a point x in the feasible region
      ii. **Check** if x is valid and if so add the sample to S.
   (c) **End for**
3. **Main step**:
   (a) **For** $\forall\, x \in S$
      i. If x is valid then start the deterministic local search procedure $L(x)$ ast used also in algorithm 1, yielding a local minimum x^*. If $x \notin X^*$ then $X^* = X^* \cup x^*$
   (b) **End For**
4. **Decision step**: **If** the Double-box stopping rule holds **then** terminate **else** Goto Sampling step.

algorithm a point is considered to be valid if it is not too close to some already located minimum or another sample in S. The closeness with a local minimum or some other sample is guided through the so-called typical distance and the gradient criterion. Further information about the algorithm can be found in [20].

3. Experiments

3.1. Test problems

Eight test cases have been examined here along with their variations. These test cases are well established in the literature and correspond to diverse scientific fields.

Effati–Grosan—problem 1

This problem is considered also in the papers of Effati [23] and Grosan [24]. The system of equations is defined as follows:

$$\begin{aligned} f_1(x_1, x_2) &= \cos(2x_1) - \cos(2x_2) - 0.4 \\ f_2(x_1, x_2) &= 2(x_2 - x_1) + \sin(2x_2) - \sin(2x_1) - 1.2 \end{aligned} \tag{4}$$

where $-a \le x_i \le a$ with $a = 2$ or $a = 10$ or $a = 100$. The total number of roots is unspecified in the literature.

Effati–Grosan—problem 2

This is another problem considered by Effati and Grosan and it is given by the following equations:

$$\begin{aligned} f_1(x_1, x_2) &= e^{x_1} + x_1x_2 - 1 \\ f_2(x_1, x_2) &= \sin(x_1x_2) + x_1 + x_2 - 1 \end{aligned} \tag{5}$$

where $-a \le x_i \le a$ with $a = 2$ or $a = 10$ or $a = 100$. The total number of roots is unspecified in the literature.

Table 1
The values for the angles $\psi_i,\ \phi_i,\ i = 0, 1, 2, 3$ for the steering problem.

i	ψ_i	ϕ_i
0	1.3954170041747090114	1.7461756494150842271
1	1.7444828545735749268	2.0364691127919609051
2	2.0656234369405315689	2.2390977868265978920
3	2.4600678478912500533	2.4600678409809344550

Reactor problem

This problem deals with a model of two continuous nonadiabatic stirred tank reactors and it is described in [25–27]. The problem is given by the following equations:

$$f_1(x_1, x_2) = (1 - R)\left(\frac{D}{10(1+\beta_1)} - x_1\right)\exp\left(\frac{10x_1}{1 + \frac{10x_1}{\gamma}}\right) - x_1$$

$$f_2(x_1, x_2) = x_1 - (1+\beta_2)x_2 + (1 - R)\left(\frac{D}{10} - \beta_1 x_1 - (1+\beta_2)x_2\right)\exp\left(\frac{10x_2}{1 + \frac{10x_2}{\gamma}}\right) \tag{6}$$

where $x_i \in [0, 1]$ and $\gamma = 1000,\ D = 22,\ \beta_1 = 2,\ \beta_2 = 2$. The parameter R takes the values: 0.935, 0.940, 0.945, 0.950, 0.955, 0.960, 0.965, 0.965, 0.970, 0.975, 0.980, 0.985, 0.990 and 0.995. The number of roots depends on the value of parameter R and it varies from 1 to 7.

Steering problem

This is a kinematic synthesis problem for automotive steering described in [28–30] and it is described by the following equations for $i = 1, 2, 3$

$$\begin{aligned} G_i(\psi_i, \phi_i) = {} & (E_i(x_2 \sin(\psi_i)) - F_i(x_2 \sin(\phi_i) - x_3))^2 + (F_i(1 + x_2\cos(\phi_i)) - E_i(x_2\cos(\psi_i) - 1))^2 \\ & - ((1 + x_2\cos(\phi_i))(x_2\sin(\psi_i) - x_3)x_1 - (x_2\sin(\phi_i) - x_3)(x_2\cos(\psi_i) - x_3)x_1)^2 \end{aligned}$$

with

$$E_i = x_2(\cos(\phi_i) - \cos(\phi_0)) - x_2x_3(\sin(\phi_i) - \sin(\phi_0)) - (x_2\sin(\phi_i) - x_3)x_1$$

and

$$F_i = -x_2\cos(\psi_i) - x_2x_3\sin(\psi_i) + x_2\cos(\psi_0) + x_1x_3 + (x_3 - x_1)x_2\sin(\psi_0)$$

and $x \in [0.06, 1]^3$. The values for the angles ϕ_i and ψ_i are shown in Table 1. For these values of the angles the systems has two roots.

Merlet problem

This problem was found in [30] and it is a system of two equations:

$$\begin{aligned} f_1(x_1, x_2) &= -\sin(x_1)\cos(x_2) - 2\cos(x_1)\sin(x_2) \\ f_2(x_1, x_2) &= -\cos(x_1)\sin(x_2) - 2\sin(x_1)\cos(x_2) \end{aligned} \tag{7}$$

with $x \in [0, 2\pi]^2$. The system has 13 roots in the specified domain.

Floudas problem

This problem is defined in [26] and it is given by the following equations:

$$f_1(x_1, x_2) = 0.5\sin(x_1x_2) - 0.25\frac{x_2}{\pi} - 0.5x_1$$

$$f_2(x_1, x_2) = \left(1 - \frac{0.25}{\pi}\right)(\exp(2x_1) - e) + e\frac{x_2}{\pi} - 2ex_1 \tag{8}$$

with $x_1 \in [0.25, 1]$ and $x_2 \in [1.5, 2\pi]$. The system has two roots in the given domain.

Yamamutra problem

The problem is considered in [32] and in [33] and it is defined by:

$$x_i - \frac{1}{2n}\left(\sum_{j=1}^{n} x_j^3 + i\right) = 0, \quad i = 1, 2, \ldots, n.$$

Table 2
Results for the Effati–Grosan problem 1.

a	ROOTS	MS1	MS2	MS3	MINF
2	1	3659(0.03)	3659(0.03)	4009(0.04)	754(0.01)
10	13	8874(0.07)	5299(0.04)	19 790(0.15)	5398(0.10)
100	127	125 349(1.02)	72 552(0.59)	164 932(1.35)	106 824(1.45)

Table 3
Results for the Effati–Grosan problem 2.

a	ROOTS	MS1	MS2	MS3	MINF
2	1	4026(0.03)	4026(0.03)	4849(0.04)	616(0.01)
10	1	5459(0.04)	5459(0.04)	6435(0.05)	715(0.01)
100	1	8361(0.06)	8361(0.06)	9837(0.07)	1135(0.01)

Table 4
Results for the reactor problem.

R	ROOTS	MS1	MS2	MS3	MINF
0.935	1	5445(0.04)	6494(0.05)	5864(0.05)	539(0.01)
0.940	1	5589(0.05)	6678(0.05)	5973(0.05)	570(0.01)
0.945	3	7817(0.06)	12502(0.10)	9469(0.08)	591(0.01)
0.950	5	7359(0.06)	8714(0.07)	9522(0.08)	2417(0.04)
0.955	5	5919(0.05)	7169(0.06)	8922(0.08)	1781(0.03)
0.960	7	5394(0.05)	5565(0.05)	11 092(0.09)	2469(0.05)
0.965	5	5448(0.05)	18 457(0.15)	8090(0.07)	1334(0.03)
0.970	5	5075(0.04)	12 361(0.11)	7490(0.06)	1020(0.02)
0.975	5	4519(0.04)	6724(0.06)	6859(0.06)	1089(0.03)
0.980	5	4786(0.04)	9014(0.08)	7432(0.06)	1137(0.03)
0.985	5	5369(0.05)	7861(0.07)	6994(0.06)	1456(0.03)
0.990	1	4106(0.04)	4106(0.04)	4620(0.04)	357(0.01)
0.995	1	3863(0.03)	3863(0.03)	4718(0.04)	296(0.01)

Table 5
Results for problems Steering, Merlet and Floudas.

Problem	ROOTS	MS1	MS2	MS3	MINF
Steering	2	11 941(0.44)	39 758(1.62)	12 886(0.47)	604(0.03)
Merlet	13	4605(0.05)	3297(0.03)	8033(0.08)	260(0.03)
Floudas	2	4273(0.03)	69 627(0.53)	4273(0.03)	1259(0.02)

Bratu problem

The problem is considered in [34] and it is defined by:

$$x_{i-1} - 2x_i + x_{i+1} + h^2 \exp(x_i) = 0, \quad i = 1, 2, \ldots, n$$

where $x_0 = x_{n+1} = 0$ and $h = \frac{1}{n+1}$.

3.2. Results

All the methods were run 30 times using different seeds for the random generator each time and averages were taken. All the experiments were performed on a Intel core duo processor equipped with 2 GB ram running Ubuntu Linux 8.04. The sample size for each method (parameter N in Algorithms 1 and 2) was set to 20 in all experiments. Results are presented in Tables 2–5. In all tables the numbers in parentheses denote the average time required. The column ROOTS denotes the average number of roots found, the column MS1 denotes the average number of function calls using the Multistart method with Double-box termination check, the column MS2 denotes the average number of function calls using the Multistart method with Observables termination check, the column MS3 denotes the average number of function calls using the Multistart method with Expected minimizer stopping rule and the column MINF denotes the average number of function calls for the Minfinder method. The local search method used in the experiments was a BFGS variant due to Powell [31]. Both Multistart and Minfinder have managed to find the same number of roots in all experiments, but they differ dramatically in the number of function evaluations.

In Tables 2 and 3 the results from the application of Multistart and Minfinder to the problems Effati–Grosan problem 1 and problem 2 respectively are listed. The column a denotes the parameter a for the left and right bounds of variables x_1, x_2 i.e. $-a \le x_i \le a$. Both Multistart and Minfinder managed to find the same number of roots for the system of equations, which means that the same roots were found for the parameter a. As it can be seen from Tables 2 and 3 a 10%–80% gain in speed

Table 6
Results for the Yamamura problem.

N	ROOTS	MS1	MS2	MS3	MINF
10	3	15 752(0.36)	192 644(4.47)	16 220(0.37)	10 022(0.25)
20	3	23 234(1.46)	117 262(7.27)	24 049(1.51)	22 721(1.48)
30	3	28 251(3.53)	145 679(17.89)	28 796(3.61)	26 865(3.44)
40	3	32 583(6.89)	167 246(37.55)	34 466(7.29)	31 161(6.71)

Table 7
Results for the Bratu problem.

N	ROOTS	MS1	MS2	MS3	MINF
10	2	13 909(0.31)	97 035(2.13)	13 909(0.31)	6654(0.17)
20	2	24 744(1.30)	25 196(1.33)	25 834(1.36)	15 139(0.84)
30	2	37 799(3.56)	145 229(24.31)	36 728(3.39)	23 932(2.88)

has been achieved using the Minfinder optimization procedure relatively to Multistart. In contrast to the methodologies founded in the literature that concentrated on locating one root, our method found multiple roots (for example 127 for the first problem with $a = 100$) without *a priori* knowledge of the target number of roots. The roots discovered in [19] were also included to the set of roots founded by our methods.

In Table 4 the results from the Reactor problem are reported. The column R denotes the value of parameter R in Eq. (6). Both Multistart and Minfinder methods were able to discover the same number of roots as in [21] without *a priori* knowledge of the target number of roots. As it can be seen from Table 4 the speed gain from the application of Minfinder method was ranged from 50% to 90% against Multistart.

In Table 5 the results from the application of the methods to the problems of Steering, Merlet and Floudas are listed. As in previous cases, Multistart and Minfinder were able to discover the same number of roots as in [21]. Also, the application of Minfinder had a significant gain in speed ranged from 75% to 90%.

In Tables 6 and 7 the results from the application of the proposed methods to Yamamura and Bratu problem are listed. The column N denotes the dimension of the objective function. Both Multistart and Minfinder managed to find all the solutions of the objective problems.

4. Conclusions

The root finding problem has been successfully transformed to global optimization problem. General applied stochastic global optimization methods have been utilized to locate the global minima of the transformed problem which correspond to the roots of the systems of nonlinear equations. Results presented here from various test cases indicate that all real roots can be found without *a priori* knowledge of the number of the roots. Efficiency of global optimization methods guarantee the locating of all possible roots. It was found also that Minfinder resulted in considerable gain in speed relatively to the classical Multistart method, although both methods showed the same efficiency. The generality of the applied methodology allows its application to any system of nonlinear equations, since it does not depend on the problem formulation and does not require *a priori* knowledge of the number of the existed roots.

Future improvements of the proposed method can be finding also complex roots of systems or application to systems of complex nonlinear equations.

References

[1] K. Kowalski, K. Jankowski, Towards complete solutions to systems of nonlinear equations of many-electron theories, Physical Review Letters 81 (1998) 1195–1198.

[2] B.M. Barbashov, V.V. Nesterenko, A.M. Chervyakov, General solutions of nonlinear equations in the geometric theory of the relativistic string, Communications in Mathematical Physics 84 (1982) 471–481.

[3] S.R. Bickham, S.A. Kiselev, A.J. Sievers, Stationary and moving intrinsic localized modes in one-dimensional monatomic lattices with cubic and quartic anharmonicity, Physical Review B 47 (1993) 14206–14211.

[4] A. Holstad, Numerical solution of nonlinear equations in chemical speciation calculations, Computational Geosciences 3 (1999) 229–257.

[5] K. Argyros, On the solution of undetermined systems of nonlinear equations in Euclidean spaces, Pure Mathematics and Applications 4 (1993) 199–209.

[6] V.S. Semenov, The method of determining all real nonmultiple roots of systems of nonlinear equations, Computational Mathematics and Mathematical Physics 47 (2007) 1428–1434.

[7] J.E. Dennis, R. Schnable, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, NJ, 1983, Mir, Moscow, 1988.

[8] V.P. Bulatov, Numerical methods for finding all real roots of systems of nonlinear equations, Zhurnal Vychislitel'noi Mathematiki i Matematicheskoi Fiziki 40 (2000) 348–355; Comput. Math. Phys. 40 (2000) 331–338.

[9] I. Alolyan, An algorithm for finding all zeros of vector functions, Bulletin of the Australian Mathematical Society 77 (2008) 353–363.

[10] J.E. Denis, On Newtons methods and nonlinear simulataneus replacements, SIAM Journal of Numerical Analysis 4 (1967) 103–108.

[11] J.M. Martinez, Algorithms for solving nonlinear systems of equations, in: E. Spedicatio (Ed.), Continuous Optimization: The State of the Art, Kluwer, 1994, pp. 81–108.

[12] P. Deuflhard, A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting, Numerische Mathematik 22 (1974) 289–315.

[13] A.H. Sherman, On Newton-iterative methods for the solution of systems of nonlinear equations, SIAM Journal on Numerical Analysis 15 (1978) 755–771.
[14] S. Bellavia, M. Macconi, B. Morini, An affine scaling trust-region approach to bound-constrained nonlinear systems, Applied Numerical Mathematics 44 (2003) 257–280.
[15] L. Lukšan, Inexact trust region method for large sparse systems of nonlinear equations, Journal of Optimization Theory and Applicationsand Applications 81 (1994) 569–590.
[16] R.B. Schnabel, P.D. Frank, Tensor methods for nonlinear equations, SIAM Journal on Numerical Analysis 21 (1994) 815–843.
[17] A. Bouaricha, R.B. Schnabel, Tensor methods for large sparse systems of nonlinear equations, Mathematical Programming 82 (1998) 377–400.
[18] C.L. Karr, B. Weck, L.M. Freeman, Solutions to systems of nonlinear equations via a genetic algorithm, Engineering Applications of Artificial Intelligence 11 (1998) 369–375.
[19] A.A. Mousa, I.M. El-Desoky, GENLS: Co-evolutionary algorithm for nonlinear systems of equations, Applied Mathematics and Computation 197 (2008) 633–642.
[20] I.G. Tsoulos, I.E. Lagaris, MinFinder: Locating all the local minima of a function, Computer Physics Communications 174 (2006) 166–179.
[21] M.J. Hirsch, P.M. Pardalos, M.G.C. Resende, Solving systems of nonlinear equations with continuous GRASP, Nonlinear Analysis: Real World Applications 10 (2009) 2000–2006.
[22] I.E. Lagaris, I.G. Tsoulos, Stopping rules for box-constrained stochastic global optimization, Applied Mathematics and Computation 197 (2008) 622–632.
[23] S. Effati, A.R. Nazemi, A new method for solving a system of the nonlinear equations, Applied Mathematics and Computation 168 (2005) 877–894.
[24] C. Crosan, A. Abraham, T.G. Chang, D.H. Kim, Solving nonlinear equation systems using an evolutionary multiobjective multiobjective optimization approach, in: 5th Mexican International Conference on Artificial Intelligence, MICAI'06, Mexico, in: Lecture Notes in Computer Science, Springer Verlag, Germany, 2006.
[25] M. Kubicek, H. Hofmann, V. Hlavacek, J. Sinkule, Multiplicity and stability in a sequence of two nonadiabatic nonisothermal CSTR, Chemical Engineering Sciences 35 (1980) 987–996.
[26] C.A. Floudas, P.M. Pardalos, C. Adjiman, W. Esposito, Z. Gumus, S. Harding, J. Klepeis, C. Meyer, C. Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer Acedemic Publishers, Dordrecht, 1999.
[27] C.A. Floudas, Recent advances in global optimization for process synthesis, design and control: Enclosure of all solutions, Computers and Chemical Engineering 23 (1999) 963–973.
[28] S. Pramanik, Kinematic synthesis of a six-member mechanism for automotive steering, ASME Journal of Mechanical Design 124 (2002) 642–645.
[29] Wikipedia, Ackerman Steering Geometry, http://en.wikipedia.org/wiki/Ackerman_steering_geometry, 2006.
[30] J.P. Merlet, The CORPIN examples page, http://www-sop.inria.gr/corpin/logiciels/ALIAS/Benches/benches.html, 2006.
[31] M.J.D Powell, A tolerant algorithm for linearly constrained optimization calculations, Mathematical Programming 45 (1989) 547–566.
[32] E. Allgower, K. Georg, Simplicial and continuation methods for approximating fixed points and solutions to systems of equations, SIAM Review 22 (1980) 28–85.
[33] K. Yamamura, H. Kawata, A. Tokue, Interval solution of nonlinear equations using linear programming, BIT Numerical Mathematics 38 (1998) 186–199.
[34] J.J. Moré, A collection of nonlinear model problems, in: E.L. Allgower, K. Georg (Eds.), Computational Solution of Nonlinear Systems of Equations, in: Lectures in Applied Mathematics, vol. 26, American Mathematical Society, Providence, RI, 1990, pp. 723–762.