SOLVING POLYNOMIAL SYSTEMS USING A MODIFIED LINE SEARCH APPROACH

Crina Grosan\textsuperscript{1}, Ajith Abraham\textsuperscript{2,3} and Vaclav Snasel\textsuperscript{2}

\textsuperscript{1}Department of Computer Science
Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca 400084, Romania
cgrosan@cs.ubbcluj.ro

\textsuperscript{2}Faculty of Electrical Engineering and Computer Science
VSB-Technical University of Ostrava
17. listopadu, 15/2172, Ostrava-Poruba 70833, Czech Republic
vaclav.snasil@vsb.cz

\textsuperscript{3}Machine Intelligence Research Labs
Scientific Network for Innovation and Research Excellence
Auburn, Washington, USA
ajith.abraham@ieee.org

Received August 2010; revised December 2010

Abstract. This paper proposes a modified line search technique for solving systems of complex nonlinear equations. Line search is a widely used iterative global search method. Since optimization strategies have been (and continue to be) successfully used for solving systems of nonlinear equations, the system is reduced to a one-dimensional equation system for optimization purpose. The proposed line search procedure incorporates a re-start technique, which makes use of derivatives to reduce the search space and to re-generate thereafter the starting points in between the new ranges. Several well known applications such as internal arithmetic benchmark, kinematics, neuropsychology, combustion, chemical equilibrium and economics application are considered for testing the performances of the proposed approach. To validate the strength of the proposed approach, systems having between 5 and 20 equations are considered. Results are compared with an evolutionary algorithm approach, which transforms the problem into a multi-objective optimization problem. Empirical results reveal that the proposed approach is able to deal with high dimensional equations systems very effectively.

1. Introduction. Polynomial systems play a fundamental role in many areas of science and engineering. In our research, we refer to systems of \( n \) polynomials in \( n \) variables. The problem of solving a system of equations is NP hard, which involves very high computational complexity due to several numerical issues [21]. There are several approaches dealing with polynomial systems reported in an impressive number of publications [20, 26, 41, 46, 47]. Manocha [24] classifies them into three main categories:

- Symbolic methods: can eliminate variables, reducing the problem to finding the roots of univariate polynomials. Such methods stem from algebraic geometry. The current algorithms and implementations are efficient only for sets of low-degree polynomial systems (no more than three or four polynomials) [48].
- Numeric methods: are based on either iterative or homotopy methods. Iterative methods, like Newton’s, are good only for local analysis and work well only with a good initial guess for each solution, which is rather difficult for applications like
intersections or geometric constraint systems. Homotopy methods, based on continuation techniques, follow paths in complex space. In theory, each path converges to a geometrically isolated solution [31].

- Geometric methods: develop some algorithms, for example, subdivision-based algorithms for intersections and ray tracing of curves and surfaces. In general, subdivision algorithms have limited applications and slow convergence [40].

This paper proposes a new approach which transforms the system of nonlinear equations into a single equation and thereafter treats it as an optimization problem. For solving the obtained optimization problem a modified line search approach is applied. The proposed approach is compared with another approach which transforms the system into a multiobjective optimization problem and solves it using an evolutionary computation technique.

The performances of the proposed approach are 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. While compared with the evolutionary approach (which in its turn outperforms other numerical methods), the current research produces more robust solutions and it is by far more effective in terms of computational cost involved.

The paper is organized as follows: in Section 2, the basic concepts about polynomial systems as well as the way in which the problem is transformed into an optimization problem are presented. Section 3 presents the Modified Line Search (MLS) approach. Section 4 briefly introduces the way in which the problem is reduced to a multiobjective optimization problem and the evolutionary technique applied to solve it. Section 5 is dedicated to numerical experiments. 12 instances from interval arithmetic, economics, kinematics, chemistry, combustion and neurophysiology are considered. Section 6 presents conclusions of the work and future research ideas.

2. Basic Concepts. A polynomial of degree $d$ in one variable (denoted by $x$) is a function of the form

$$a_0x^d + a_1x^{d-1} + \ldots + a_{d-1}x + a_d$$

where $a_0, \ldots, a_d$ are the coefficients and the integer powers of $x$, namely $1, x, x^2, \ldots, x^d$, are monomials. In science and engineering, such functions usually have coefficients that are real numbers although sometimes they may be complex [41].

**Definition 2.1. (Polynomial)** A function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ in $n$ variables $x = (x_1, \ldots, x_n)$ is a polynomial if it can be expressed as a sum of terms, where each term is the product of a coefficient and a monomial, each coefficient is a real number, and each monomial is a product of variables raised to nonnegative integer powers.

A system of polynomials is defined as:

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

where $x = (x_1, x_2, \ldots, x_n)$ and $f_1, \ldots, f_n$ are polynomials in the space of all real valued continuous functions on $\Omega = \prod_{i=1}^{n} [a_i, b_i] \subset \mathbb{R}^n$. 
Finding a solution for a polynomial system of equations $f(x)$ involves finding a solution such that every equation in the nonlinear system is 0:

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

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 [35, 37]. These methods use reduction to simpler one-dimensional nonlinear equations for the components $f_1, f_2, \ldots, f_n$ [18]. In a strategy based on trust regions [25], 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, sometimes referred to as continuation methods [22, 25, 34]. 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 by finding 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 [22]. The Dimension Reducing method, the Modified Reducing Dimension Method and the Perturbed Dimension Reducing Method [13, 14, 15, 16, 17] are also methods for numerical solutions of systems of nonlinear equations which incorporate Newton and nonlinear Successive Over Relaxation (SOR) algorithms [35] and use reduction to simple one dimensional nonlinear equations (but they converge quadratically).

In the approach proposed in [32], the system of equations is transformed into a constraint optimization problem. At each step some equations which 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 $S_1$ and $S_2$, where $S_2$ denotes the complement $\{1, 2, \ldots, n\} \setminus S_1$. Then, the problem is:

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

The system is reduced to the same form in the approach used in [33].

The optimization problem obtained in [22] by transforming the systems of equations is similar to the one proposed in [32] and considers the equation given by the sum of squared components $f_1, f_2, \ldots, f_n$. The approach used in [19] transforms the systems of equations into a multiobjective optimization problem, each equation representing an objective and then standard Pareto dominance relationship is used in order to determine the real solutions of the system.

The proposed approach transforms the equation systems into an optimization problem using the model presented in [32] as follows:

$$
\text{minimize } \sum_{i=1}^{n} f_i^2(x)
$$

3. **Modified Line Search.** Line search is a well established optimization technique. The modification proposed in this paper for the standard line search technique refers to step setting and also the incorporation of a re-start approach. To fine tune the performance,
the first partial derivatives of the function to optimize are also made use of. The proposed three modifications are summarized below and are described in detail in the subsequent sections:

1. The first modification refers to the inclusion of multi start principle within the search process.
2. The second modification is related to the setting of the direction and step.
3. The third modification refers to the re-starting of the line search method.

After a given number of iterations, the process is restarted by reconsidering other arbitrary starting point (or other multiple arbitrary starting points), which are generated by taking into account the results obtained at the end of previous set of iterations.

In the following subsections, the modified line search algorithm is presented in a structured form.

3.1. Generation of the starting points. It is known that the line search techniques use a starting point. There are also versions which allow the usage of multiple points and the search starts separately from each of these points. In the proposed approach, multiple arbitrary starting points are used. Each point is randomly generated over the definition domain.

For a function of \(n\) variables and the domain of definition given by:

\[
[min_1, max_1] \times [min_2, max_2] \times \ldots \times [min_n, max_n]
\]

where \( [min_i, max_i] \) is the domain of \(i^{th}\) variable, the procedure for generating the starting points \(x\) between the considered limits is given in Algorithm 1:

```
Algorithm 1 Generate starting points()
for i = 1 to Number of arbitrary starting points
    for j = 1 to No of variables
        \(x_{ij} = \text{min}_j + \text{random} \times (\text{max}_j - \text{min}_j); \)
    endfor
endfor
```

The \textit{random} function generates an arbitrary number between \([0,1]\).

3.2. Direction and step settings. Several experiments were conducted in order to set an adequate value for the direction. We used the standard value \(+1\) or \(-1\) and, for some functions the value \(-1\) was favorable to obtain good performance. We also performed some experiments by setting the direction value to be a random number between \(0\) and \(1\). Using the random number helped to obtain overall very good performance for the entire set of considered test functions. So, either of these values (the random value and the value \(-1\)) may be used for better performance. Experiment results reported in this paper used the random value.

The step is set as follows:

\[
\alpha_k = 2 + \frac{3}{2^{k+1}},
\]

where \(k\) refers to the iteration number.

The \textit{Line-search()} technique may be written as in Algorithm 2.
Algorithm 2 Line_search()

Set \( k = 1 \) (Number of iterations)

Repeat
    for \( i = 1 \) to Number of starting points
        for \( j = 1 \) to Number of variables
            \( p_k = \text{random}; \)
            \( \alpha_k = 2 + \frac{3}{2^2 + 1} \)
            \( x_{ij}^{k+1} = x_{ij}^k + p_k \cdot \alpha_k \)
        endfor
        if \( f(x_i^{k+1}) > f(x_i^k) \)
            then \( x_i^{k+1} = x_i^k \)
        Endfor
    \( k = k + 1 \)
Until \( k = \) Number of iterations (apriori known).

Remark 3.1. The condition:

\[
\text{if } f(x_i^{k+1}) > f(x_i^k) \text{ then } x_i^{k+1} = x_i^k
\]

allows us to move to the new generated point only if there is an improvement in the quality of the function.

Remark 3.2. Number of iterations for which line search is applied is apriori known and is usually a small number. In our experiments, we set the number of these iterations to 10. There is a single benchmark for which only one iteration was used (which means the boundaries are updated and the process is re-started after each iteration).

Remark 3.3. When restarting the line search method (after the insertion of the re-start technique), the iteration count starts again from 1 (this should not subsequently be related to the value of \( \alpha \) after the first set of iterations).

Remark 3.4. Several experiments were performed to set a value for the step, starting with random values (until a point is reached for which the objective function is getting a better value); using a starting value for the step and generating random numbers with Gaussian distribution around this number, etc. There are other ways to set the step, but the formula presented seems to perform well.

3.3. Re-start insertion. With the best result obtained in the previous set of iterations, the following steps are used to restart the algorithm:

- Among all the considered points, the solution for which the objective function obtained the best value is selected. If there are several such solutions, one of them is randomly selected. This solution will be a multi-dimensional point in the search space and denoted by \( x \) for an easier reference.
- For each dimension \( i \) of the point \( x \), the first partial derivative with respect to this dimension is calculated. This means the gradient of the objective function is calculated which is denoted by \( g \). Taking this into account, the bounds of the definition domain for each dimension is re-calculated as follows:
  \[
  \text{if } g_i = \frac{\partial f}{\partial x_i} > 0 \text{ then } \max_i = x_i; \\
  \text{if } g_i = \frac{\partial f}{\partial x_i} < 0 \text{ then } \min_i = x_i.
  \]
- The search process is re-started by re-initializing a new set of arbitrary points (using Generate starting points() procedure) but between the newly obtained boundaries (between the new \( \max_i \) or new \( \min_i \)).
The pseudo code of the Re-start technique is given in Algorithm 3 ($g$ denotes the gradient of $f$).

**Algorithm 3** Re_start()

Calculate the solution (out of the entire set of points) for which the value of the function is minimum.

Let $x^\#$ be minimum obtained at the current moment of the search process.

For $i = 1$ to Number of dimensions

- if $g_i(x^\#) > 0$ then $\max_i = x^\#_i$
- if $g_i(x^\#) < 0$ then $\min_i = x^\#_i$

endfor

3.4. **Modified line search with re-start (MLS) procedure.** The Line Search method presented in the previous subsections combined with the re-start technique described above is expressed using the pseudo code as illustrated in Algorithm 4.

**Algorithm 4** Modified Line Search with Re-Start (MLS).

Set $t = 1$;

Repeat

- Generate starting points (max, min);
- Line_search($t$);
- Re_start (new values for max$_i$ and/or min$_i$ for each dimension will be obtained);
- $t = t + 1$;

Until $t = \text{Number of applications of the re-start technique (apriori known)}$.

Select the solution $x^*$ for which the value of the objective function is minimum.

Print $x^*$.

4. **Multiobjective Evolutionary Algorithm Approach.** Evolutionary computation offers practical advantages to the researcher facing difficult optimization problems. These advantages are multi-fold, including the simplicity of the approach, its robust response to changing circumstance, its flexibility and many other facets. A population of candidate solutions (for the optimization task to be solved) is initialized. New solutions are created by applying reproduction operators (mutation and/or crossover). The fitness (how good the solutions are) of the resulting solutions are evaluated and a suitable selection strategy is then applied to determine which solutions will be maintained into the next generation.

An evolutionary technique is proposed in [19] for solving the multiobjective optimization problem obtained by transforming the system of equations. Some basic definitions of a multiobjective optimization problem and the optimality concept of the solutions are presented in [44].

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

$$f_i : \Omega \to \mathbb{R}, \quad i = 1, 2, \ldots, n$$

where $\Omega \subset \mathbb{R}^m$.

The multiobjective optimization problem is defined as:

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

For deciding whether a solution is better than another solution or not, the following relationship between solutions might be used.
Definition 4.1. (Pareto dominance) Consider a maximization problem. Let \( x, y \) be two decision vectors (solutions) from \( \Omega \).

Solution \( x \) dominates \( y \) (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’ and ‘functional-efficient’ solutions. The solution \( x^0 \) is ideal if all objectives have their optimum in a common point \( x^0 \).

Definition 4.2. (Pareto front) The images of the Pareto optimum points in the criteria space are called the Pareto front.

In the approach used in [19], the system of equations \((P)\) is transformed into a multiobjective optimization problem. Each equation is considered as an objective function. The goal of this optimization function is to minimize the differences (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 the system:

\[
(P') \left\{ \begin{array}{l}
\text{abs}(f_1(x_1, x_2, \ldots, x_n)) \\
\text{abs}(f_2(x_1, x_2, \ldots, x_n)) \\
\vdots \\
\text{abs}(f_n(x_1, x_2, \ldots, x_n))
\end{array} \right.
\]

In order 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 non-dominated 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 which will become dominated. When the size of this set is overloaded some of the solutions are removed.

The population of the next iteration is 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 5. Termination criteria of Algorithm 5 refers to a specified number of iterations.

Algorithm 5 Evolutionary Approach for solving polynomial systems as multiobjective optimization problems.

\textbf{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 cardinality of \( E \) exceeds the allowed size reduce the number of solutions in respect to the sum of 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 individual equal to the size of \( P(t) \) are obtained.

Let \( Q(t) \) be the set obtained from the best between the solutions which 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 \( Q(t) \).

Step 2.3. Update \( E \) with the nondominated individuals from \( P(t) \cup Q(t) \) 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 termination criteria is reached then go to Step 4
else go to Step 2.

Step 4.

Print \( E \).

5. Experimental Results. This section reports several experimental results for a wide class of well known applications. The problems are taken from interval analysis, kinematics, economics, combustion, chemistry and neuropsychology. 12 systems having between 5 and 20 equations are considered. Results obtained by the modified line search are compared to the results obtained by the evolutionary approach. The following three criteria are used to compare performance results:

- The Euclidian norm (denoted by \( En \)) of the vector \( f \) which was suggested in 1965 by Broyden [3] and given by:

\[
En = \sqrt{\sum_{i=1}^{n} f_i^2(x)}
\]

- Running time.
- The ratio between the number of solutions obtained by each method which are dominated by the solutions obtained by the other method and the number of solutions obtained by each method which dominates solution obtained by the other method (in terms of Pareto dominance).

More specifically, suppose the set of solutions obtained by the two methods are \( S_1 \) and \( S_2 \) respectively. Then the dominance ratio for the first technique (denoted by \( D_1 \)) is given by:

\[
D_1 = \frac{\text{dominated}(S_2, S_1)}{\text{dominate}(S_1, S_2)}
\]

where \( \text{dominated}(S_2, S_1) \) returns the number of solutions from the set \( S_1 \) which are dominated by solutions from the set \( S_2 \) and \( \text{dominate}(S_1, S_2) \) returns the number of solutions from the set \( S_1 \) which dominate solutions from the set \( S_2 \). We are not effectively computing the value of this ratio. Sometimes the denominator can be 0. We are just showing this
ratio with respect to the nondominance between two sets of solutions. Also, the value of $D_2 = \frac{1}{D_1}$.

This third criteria for comparison is mainly used because results obtained by the evolutionary approach are reported in terms of nondominance.

The benchmark problems considered for performance evaluation (which are described in detail in the following subsections) are given in Table 1. We consider and compare a set of 8 solutions obtained by each approach.

**Table 1. The 12 benchmarks used in experiments**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Number of variables</th>
<th>Range</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Interval i1</td>
<td>10</td>
<td>-2, 2</td>
<td>21, 23, 28</td>
</tr>
<tr>
<td>2. Interval i2</td>
<td>20</td>
<td>-1, 2</td>
<td>21, 23, 28</td>
</tr>
<tr>
<td>3. Interval i3</td>
<td>20</td>
<td>-2, 2</td>
<td>21, 23, 28</td>
</tr>
<tr>
<td>4. Interval i4</td>
<td>10</td>
<td>-1, 1</td>
<td>21, 23, 28</td>
</tr>
<tr>
<td>5. Interval i5</td>
<td>10</td>
<td>-1, 1</td>
<td>21, 23, 28</td>
</tr>
<tr>
<td>6. Neurophysiology application</td>
<td>6</td>
<td>-10, 10</td>
<td>45</td>
</tr>
<tr>
<td>7. Chemical equilibrium</td>
<td>5</td>
<td>-10, 10</td>
<td>29</td>
</tr>
<tr>
<td>8. Kinematics kin1</td>
<td>12</td>
<td>-10, 10</td>
<td>23</td>
</tr>
<tr>
<td>9. Kinematics kin2</td>
<td>8</td>
<td>-10, 10</td>
<td>23</td>
</tr>
<tr>
<td>10. Combustion application</td>
<td>10</td>
<td>-10, 10</td>
<td>29</td>
</tr>
<tr>
<td>11. Economics e1</td>
<td>10</td>
<td>-10, 10</td>
<td>29</td>
</tr>
<tr>
<td>12. Economics e2</td>
<td>20</td>
<td>-10, 10</td>
<td>29</td>
</tr>
</tbody>
</table>

Parameters used by the MLS and Evolutionary Approach are listed in Table 2.

**Table 2. MLS and evolutionary approach parameter settings**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>MLS</th>
<th>i1</th>
<th>i3</th>
<th>i4, i5</th>
<th>chemical</th>
<th>kin1</th>
<th>kin2</th>
<th>combustion</th>
<th>e1</th>
<th>e2</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of starting arbitrary points</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>No of re-starts (reinitializations)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>No of iterations per each restarting phase</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Evolutionary approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Population size</td>
<td>300</td>
<td>500</td>
<td>300</td>
<td>300</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>300</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>External set size</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Number of generations</td>
<td>250</td>
<td>300</td>
<td>250</td>
<td>250</td>
<td>500</td>
<td>500</td>
<td>1000</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Sigma (for mutation)</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Tournament size</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

5.1. Interval arithmetic benchmarks. We consider five interval arithmetic standard benchmarks [21, 23, 28].
5.1.1. Benchmark i1. The benchmark i1 consists of the following system of equations:

\[
\begin{align*}
0 &= x_1 - 0.25428722 - 0.18324757 x_4 x_5 x_9 \\
0 &= x_2 - 0.37842197 - 0.16275449 x_1 x_5 x_6 \\
0 &= x_3 - 0.27162577 - 0.16955071 x_1 x_2 x_{10} \\
0 &= x_4 - 0.19807914 - 0.15585316 x_7 x_1 x_6 \\
0 &= x_5 - 0.44166728 - 0.19950920 x_4 x_6 x_3 \\
0 &= x_6 - 0.14654113 - 0.18922793 x_8 x_5 x_{10} \\
0 &= x_7 - 0.42397161 - 0.21180486 x_2 x_5 x_8 \\
0 &= x_8 - 0.07056438 - 0.17081208 x_1 x_7 x_6 \\
0 &= x_9 - 0.34504906 - 0.19612740 x_10 x_6 x_8 \\
0 &= x_{10} - 0.42651102 - 0.21466544 x_4 x_8 x_1 
\end{align*}
\]

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 3. The time displayed is in milliseconds.

<table>
<thead>
<tr>
<th>Solution</th>
<th>En</th>
<th>$D_{\text{MLS}}$</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>En</th>
<th>$D_{\text{EA}}$</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.19190</td>
<td>1/1</td>
<td>516</td>
<td>Sol. 1</td>
<td>0.74686</td>
<td>1/1</td>
<td>39,077</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.21902</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>0.68843</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.22084</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>0.79977</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.19419</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>0.80754</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.18906</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>0.88129</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.21086</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>0.81442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.21813</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>0.82571</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.22125</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>0.81332</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The convergence of all the 8 solutions obtained by MLS is depicted in Figure 1(a). The comparison of En obtained by the MLS and Evolutionary Approach is presented in Figure 2(a).

As evident from the results presented, MLS obtained better results in terms of running time and Euclidian norm. For this example, $D_{\text{MLS}}$ and $D_{\text{EA}}$ have the same value: 1/1.

5.1.2. Benchmark i2. Benchmark i2 consists of the set of equations given in Figure 3.

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 4. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 1(b). The comparison of En obtained by the MLS and Evolutionary Approach is presented in Figure 2(b).

As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. $D_{\text{MLS}}$ and $D_{\text{EA}}$ have the same value: 0/0.

5.1.3. Benchmark i3. Benchmark i3 has the same equations as i2 but has initial intervals $[-2, 2]$.

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 5. The time displayed is in milliseconds.

The convergence of all 8 solutions obtained by MLS is depicted in Figure 1(c). The comparison of En obtained by the MLS and Evolutionary Approach is presented in Figure 2(c).

As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. For this example, $D_{\text{MLS}}$ and $D_{\text{EA}}$ have the same value: 0/0.
Figure 1. Convergence of the 8 solutions obtained by MLS for interval benchmarks: (a)-benchmark i1, (b)-benchmark i2, (c)-benchmark i3, (d)-benchmark i4 and (e)-benchmark i5.
Figure 2. Comparison of euclidian norm for MLS and EA for interval benchmarks: (a)-benchmark i1, (b)-benchmark i2, (c)-benchmark i3, (d)-benchmark i4 and (e)-benchmark i5

5.1.4. Benchmark i4. Benchmark i4 consists of the equations given below:

\[
\begin{align*}
0 &= x_1^2 - 0.25428722 - 0.18324757x_3^2 - 0.2^2x_9^2 \\
0 &= x_2^2 - 0.37842197 - 0.16275449x_1^2 - 0.2^2x_6^2 \\
0 &= x_3^2 - 0.27162577 - 0.16955071x_2^2 - 0.2^2x_{10}^2 \\
0 &= x_4^2 - 0.19807914 - 0.15585316x_7^2 - 0.2^2x_5^2 \\
0 &= x_5^2 - 0.44166728 - 0.19950920x_3^2 - 0.2^2x_3^2 \\
0 &= x_6^2 - 0.14654113 - 0.18922793x_4^2 - 0.2^2x_{10}^2 \\
0 &= x_7^2 - 0.42937161 - 0.21180486x_5^2 - 0.2^2x_8^2 \\
0 &= x_8^2 - 0.07056438 - 0.19612740x_7^2 - 0.2^2x_6^2 \\
0 &= x_9^2 - 0.34504906 - 0.19612740x_8^2 - 0.2^2x_6^2 \\
0 &= x_{10}^2 - 0.42651102 - 0.21466544x_9^2 - 0.2^2x_1^2
\end{align*}
\]
\[
\begin{align*}
0 &= x_1 - 0.24863995 - 0.19594124 \cdot x_2 \cdot x_9 \\
0 &= x_2 - 0.87528857 - 0.056127619 \cdot x_3 \cdot x_7 \\
0 &= x_3 - 0.23939835 - 0.20177810 \cdot x_4 \cdot x_8 \\
0 &= x_4 - 0.47620128 - 0.16497518 \cdot x_5 \cdot x_6 \\
0 &= x_5 - 0.24711044 - 0.20198778 \cdot x_9 \cdot x_10 \\
0 &= x_6 - 0.33565227 - 0.18724045 \cdot x_7 \cdot x_1 x_0 \\
0 &= x_7 - 0.13128974 - 0.13843421 \cdot x_1 \cdot x_3 \cdot x_5 \\
0 &= x_8 - 0.45937304 - 0.18180253 \cdot x_2 \cdot x_9 \cdot x_10 \\
0 &= x_9 - 0.46896600 - 0.21241045 \cdot x_3 \cdot x_7 \\
0 &= x_{10} - 0.57596885 \cdot 0.16522613 \cdot x_2 \cdot x_3 \\
0 &= x_{11} - 0.56896263 - 0.17221383 \cdot x_7 \cdot x_8 \\
0 &= x_{12} - 0.70561396 - 0.23556251 \cdot x_4 \cdot x_1 \cdot x_4 \\
0 &= x_{13} - 0.59642512 - 0.24457135 \cdot x_2 \cdot x_5 \cdot x_9 \\
0 &= x_{14} - 0.46588640 - 0.21790395 \cdot x_3 \cdot x_5 \cdot x_10 \\
0 &= x_{15} - 0.10607114 - 0.20920602 \cdot x_9 \cdot x_{10} \\
0 &= x_{16} - 0.26516898 - 0.2137773 \cdot x_9 \cdot x_{10} \\
0 &= x_{17} - 0.20436664 - 0.19838792 \cdot x_9 \cdot x_{10} \\
0 &= x_{18} - 0.56003141 - 0.18114505 \cdot x_9 \cdot x_{10} \\
0 &= x_{19} - 0.92894617 - 0.04417537 \cdot x_9 \cdot x_{10} \\
0 &= x_{20} - 0.57001682 - 0.17949149 \cdot x_9 \cdot x_{10}
\end{align*}
\]

**Figure 3.** Benchmark i2

**Table 4.** Comparison of results obtained by MLS and the evolutionary approach for the benchmark i2

<table>
<thead>
<tr>
<th>Solution</th>
<th>( E_n )</th>
<th>( D_{MLS} )</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>( E_n )</th>
<th>( D_{EA} )</th>
<th>Approach</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.60634</td>
<td></td>
<td></td>
<td>Sol. 1</td>
<td>2.20242</td>
<td></td>
<td></td>
<td>38,750</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.65069</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>2.25937</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.65508</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>2.17129</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.66470</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>2.27246</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.66435</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>2.21490</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.67599</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>1.85480</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.69210</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>2.09113</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.68729</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>2.06740</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.** Comparison of the results obtained for the benchmark i3

<table>
<thead>
<tr>
<th>Solution</th>
<th>( E_n )</th>
<th>( D_{MLS} )</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>( E_n )</th>
<th>( D_{EA} )</th>
<th>Approach</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.61134</td>
<td></td>
<td></td>
<td>Sol. 1</td>
<td>2.20300</td>
<td></td>
<td></td>
<td>44,655</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.63608</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>2.20944</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.65165</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>2.20199</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.65716</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>2.17253</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.67047</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>2.18607</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.67475</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>2.24221</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.68942</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>2.10679</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.70766</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>2.38613</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6. Comparison of the results obtained by MLS and the evolutionary approach for the benchmark i4

<table>
<thead>
<tr>
<th>Solution</th>
<th>MLS</th>
<th>Ev. Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$En$</td>
<td>$D_{MLS}$</td>
</tr>
<tr>
<td>Sol. 1</td>
<td>0.207/34</td>
<td>0/6</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.21872</td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.24194</td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.26271</td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.26738</td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.27376</td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.27924</td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.27768</td>
<td></td>
</tr>
</tbody>
</table>

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 6. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 1(d). The comparison of $En$ obtained by the MLS and Evolutionary Approach is presented in Figure 2(d).

As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. The values of $D_{MLS}$ and $D_{EA}$ are 0/6 and 6/0 respectively. This means that no solutions obtained by MLS are dominated by solutions obtained by the evolutionary approach while 6 of the solutions obtained by the evolutionary approach are dominated by solutions obtained by MLS.

5.1.5. Benchmark i5. Benchmark i5 consists of the equations given below.

\[
\begin{align*}
0 &= x_1^2 - 0.25428722 - 0.18324757x_3^3x_9^3 + x_3^4x_9^7 \\
0 &= x_2^2 - 0.37142197 - 0.16275449x_1^3x_{10}^3 + x_1^4x_{10}^6 \\
0 &= x_3^2 - 0.27162577 - 0.16955071x_1^3x_3^3x_9^3 + x_3^4x_{10}^4 \\
0 &= x_4^2 - 0.19807914 - 0.15585316x_7^3x_6^3x_8^3 + x_6^4x_7^4 \\
0 &= x_5^2 - 0.44166728 - 0.19950920x_7^3x_6^3x_3^3 + x_6^4x_3^4 \\
0 &= x_6^2 - 0.14654113 - 0.18922793x_7^3x_6^3x_8^3 + x_8^4x_6^4 \\
0 &= x_7^2 - 0.42937161 - 0.21180486x_7^3x_5^3x_8^3 + x_5^4x_8^4 \\
0 &= x_8^2 - 0.07056438 - 0.19612740x_7^3x_5^3x_9^3 + x_9^4x_7^4 \\
0 &= x_9^2 - 0.34504006 - 0.19612740x_{10}^3x_5^3x_8^3 + x_8^4x_{10}^4 \\
0 &= x_{10}^2 - 0.42651102 - 0.21466544x_4^3x_8^3x_1^3 + x_8^4x_1^4
\end{align*}
\]

Table 7. Comparison of the results obtained by the MLS and evolutionary approach for the benchmark i5

<table>
<thead>
<tr>
<th>Solution</th>
<th>MLS</th>
<th>Ev. Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$En$</td>
<td>$D_{MLS}$</td>
</tr>
<tr>
<td>Sol. 1</td>
<td>0.23610</td>
<td>0/1</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.23830</td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.24163</td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.24384</td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.24698</td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.24618</td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.25565</td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.24702</td>
<td></td>
</tr>
</tbody>
</table>

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 7. The time displayed is in milliseconds. The convergence of all 8
solutions obtained by MLS is depicted in Figure 1(e). The comparison of the $E_n$ obtained by the MLS and Evolutionary Approach is presented in Figure 2(e). As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. For this example the values of $D_{MLS}$ and $D_{EA}$ are 0/1 and 1/0 respectively. This means that no solutions obtained by MLS are dominated by solutions obtained by the evolutionary approach while one solution obtained by the evolutionary approach is dominated by solutions obtained by MLS.

5.2. A neurophysiology application. We considered the example proposed in [45] and consisting of the equations given below:

$$\begin{align*}
    x_1^2 + x_2^2 &= 1 \\
    x_2^2 + x_4^2 &= 1 \\
    x_5x_3 + x_6x_4 &= c_1 \\
    x_5x_1 + x_6x_2 &= c_2 \\
    x_5x_1x_3 + x_6x_2x_4 &= c_3 \\
    x_5^2x_3 + x_6^2x_4 &= c_4
\end{align*}$$

The constants $c_i$ can be randomly chosen. In our experiments, we considered $c_i = 0$, $i = 1, \ldots, 4$.

**Table 8.** Comparison of the results obtained by the MLS and evolutionary approach for neurophysiology application

<table>
<thead>
<tr>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{MLS}$</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{EA}$</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.01998</td>
<td>0/6</td>
<td>922</td>
<td>Sol. 1</td>
<td>0.34755</td>
<td>6/0</td>
<td>28,906</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.05158</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>0.38236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.05608</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>0.38443</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.05769</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>0.36612</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.06200</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>0.59597</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.06544</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>0.34708</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.07451</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>0.34823</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.08732</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>0.20563</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 8. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 4. The comparison of $E_n$ obtained by the MLS and Evolutionary Approach is presented in Figure 5.

As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. For this example, the values of $D_{MLS}$ and $D_{EA}$ are 0/6 and 6/0 respectively. This means that no solutions obtained by MLS are dominated by solutions obtained by the evolutionary approach while six of the solutions obtained by the evolutionary approach are dominated by solutions obtained by MLS.

5.3. Chemical equilibrium application. We consider the chemical equilibrium system given by the following equations [27] (see also [21]):

$$\begin{align*}
    x_1x_2 + x_1 - 3x_5 &= 0 \\
    2x_1x_2 + x_1 + x_2x_3^2 + R_6x_2 - R_5x_5 + 2R_{10}x_2^2 + R_7x_2x_3 + R_9x_2x_4 &= 0 \\
    2x_2x_3^2 + 2R_5x_3^2 - 8x_5 + R_6x_3 + R_7x_2x_3 &= 0 \\
    R_6x_2x_4 + 2x_2^2 - 4R_{10}x_5 &= 0 \\
    x_1(x_2 + 1) + R_10x_2^2 + x_2x_3^2 + R_6x_2 + R_5x_3 + x_4^{-1} - 1 + R_6x_3 + R_7x_2x_3 + R_9x_2x_4 &= 0
\end{align*}$$
where
\[
\begin{align*}
R &= 10 \\
R_5 &= 0.193 \\
R_6 &= \frac{0.002567}{\sqrt{10}} \\
R_7 &= \frac{0.00348}{\sqrt{10}} \\
R_8 &= \frac{0.000137}{\sqrt{10}} \\
R_9 &= \frac{0.0002155}{\sqrt{10}} \\
R_{10} &= \frac{0.0003846}{\sqrt{10}}
\end{align*}
\]

![Figure 4](image1.png)

**Figure 4.** Convergence of the 8 solutions obtained by MLS for the neurophysiology application

![Figure 5](image2.png)

**Figure 5.** Comparison of euclidean norm obtained by MLS and the evolutionary approach for the neurophysiology application

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 9. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 6. The comparison of the \( En \) values obtained by MLS and the evolutionary approach is presented in Figure 7.

As evident from the results presented, MLS obtains better results in terms of running time and Euclidean norm. For this example, the values of \( D_{MLS} \) and \( D_{EA} \) are 0/4 and 4/0 respectively. This means that no solutions obtained by MLS are dominated by solutions obtained by the evolutionary approach while four of the solutions obtained by the evolutionary approach are dominated by solutions obtained by MLS.
SOLVING POLYNOMIAL SYSTEMS USING THE MLS APPROACH

![Graph showing convergence of solutions](image1)

**Figure 6.** Convergence of the 8 solutions obtained by MLS for the chemical equilibrium application

![Graph showing comparison of Euclidean norm](image2)

**Figure 7.** Comparison of Euclidean norm for MLS and evolutionary approach for chemical equilibrium application

**Table 9.** Comparison of the results obtained by the MLS and evolutionary approach for the chemical equilibrium application

<table>
<thead>
<tr>
<th>Solution</th>
<th>( E_n )</th>
<th>( D_{E\text{MLS}} )</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>( E_n )</th>
<th>( D_{E\text{EA}} )</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.16072</td>
<td>0/4</td>
<td>922</td>
<td>Sol. 1</td>
<td>0.50300</td>
<td>4/0</td>
<td>28,906</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.21222</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>0.65838</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.29692</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>0.61190</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.30787</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>0.50826</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.32114</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>0.33594</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.34612</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>0.40933</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.35578</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>0.37625</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.35341</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>0.53241</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.4. Kinematic applications.

5.4.1. **Kinematic application kin1.** Application kin1 comes from robotics and describes the inverse kinematics of an elbow manipulator [23]. It consists of a sparse system with 12 variables as given below.
\[
\begin{align*}
  s_2 c_5 s_6 - s_3 c_5 s_6 - s_4 c_5 s_6 + c_2 c_6 + c_3 c_6 + c_4 c_6 &= 0.4077 \\
  c_1 c_2 s_5 + c_1 c_3 s_5 + c_1 c_4 s_5 + s_1 c_5 &= 1.9115 \\
  s_2 s_5 + s_3 s_5 + s_4 s_5 &= 1.9791 \\
  s_1 c_2 + c_1 c_3 + c_1 c_4 + c_1 c_5 + c_1 c_6 &= 4.0616 \\
  s_1 c_2 + s_1 c_3 + s_1 c_4 + s_1 c_5 + s_1 c_6 &= 1.7172 \\
  s_2 + s_3 + s_4 + s_2 + s_3 + s_2 &= 3.9701 \\
  s_i^2 + c_i^2 &= 1, \quad 1 \leq i \leq 6
\end{align*}
\]

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 10. The convergence of all 8 solutions obtained by MLS is depicted in Figure 8(a). The comparison of the \( En \) values obtained by the MLS and Evolutionary Approach is presented in Figure 9(a). As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. The value of \( D_{MLS} \) is 0/8 and the value of \( D_{EA} \) is 8/0 which means that all 8 solutions obtained by the evolutionary approach are dominated.

<table>
<thead>
<tr>
<th>Solution</th>
<th>( En )</th>
<th>( D_{MLS} )</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>( En )</th>
<th>( D_{EA} )</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.64788</td>
<td></td>
<td>0/8</td>
<td>Sol. 1</td>
<td>6.96304</td>
<td></td>
<td>8/0</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.76080</td>
<td></td>
<td>937</td>
<td>Sol. 2</td>
<td>6.72439</td>
<td></td>
<td>109,327</td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.74959</td>
<td></td>
<td>354</td>
<td>Sol. 3</td>
<td>6.38505</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.83451</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>6.79585</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.83541</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>6.40160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.83830</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>6.88722</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.84435</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>6.44937</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.85331</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>6.35426</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.4.2. Kinematic application kin2. We consider the kinematic application kin2 as introduced in [29] (see also [21]) which describes the inverse position problem for a six revolute joint problem in mechanics. The equations describe a denser constraint system and are given below:

\[
\begin{align*}
  x_i^2 + x_{i+1}^2 - 1 &= 0 \\
  a_{1i} x_1 x_3 + a_{2i} x_1 x_4 + a_{3i} x_2 x_3 + a_{4i} x_2 x_4 + a_{5i} x_2 x_7 + a_{6i} x_5 x_8 + a_{7i} x_6 x_7 + a_{8i} x_6 x_8 + a_{9i} x_1 + a_{10i} x_2 + a_{11i} x_3 + a_{12i} x_4 + a_{13i} x_5 + a_{14i} x_6 + a_{15i} x_7 + a_{16i} x_8 + a_{17i} &= 0
\end{align*}
\]

The coefficients \( a_{ki}, 1 \leq k \leq 17, 1 \leq i \leq 4 \) for kin2 are given in Table 11.

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 12. The time displayed is in milliseconds.

The convergence of all 8 solutions obtained by MLS is depicted in Figure 8(b). The comparison of the \( En \) values obtained by MLS and Evolutionary Approach is presented in Figure 9(b).

As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. \( D_{MLS} \) and \( D_{EA} \) have the values 0/4 and 4/0 respectively. This means that no solutions obtained by MLS are dominated by solutions obtained by the
Figure 8. Convergence of the 8 solutions obtained by MLS for the kinematics applications: (a)-kin1 and (b)-kin2.

Figure 9. Comparison of euclidian norm for MLS and evolutionary approach for kinematics applications: (a)-kin1 and (b)-kin2.

An evolutionary approach while four of the solutions obtained by the evolutionary approach are dominated by solutions obtained by MLS.

5.5. Combustion application. We consider the combustion problem for a temperature of 3000° as proposed in [30] (see also [21]). The problem is described by the sparse system
Table 11. Coefficients $a_{kl}$ for the kinematic example kin2

<table>
<thead>
<tr>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{MLS}$</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.38881</td>
<td>0.125016350</td>
<td>-0.635550077</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.40146</td>
<td>-0.68607360</td>
<td>0.115719920</td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.43731</td>
<td>-0.11928120</td>
<td>-0.666404480</td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.47221</td>
<td>+1.343801600</td>
<td>-0.719990470</td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.49747</td>
<td>-0.000000000</td>
<td>-0.432419270</td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.49547</td>
<td>+0.400263840</td>
<td>+0.000000000</td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.50620</td>
<td>-0.300527680</td>
<td>+0.000000000</td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.49329</td>
<td>-0.386159610</td>
<td>+0.085383482</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.755266030</td>
<td>+0.000000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.542016800</td>
<td>-0.039251967</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1.091628700</td>
<td>+0.000000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.000000000</td>
<td>-0.1162908100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.049207290</td>
<td>+0.000000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+0.049207290</td>
<td>+0.013873010</td>
</tr>
</tbody>
</table>

Table 12. Comparison of the results obtained by the MLS and evolutionary approach for the kinematics application kin2

<table>
<thead>
<tr>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{MLS}$</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{EA}$</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.38881</td>
<td>0.47221</td>
<td>0/4</td>
<td>969</td>
<td>Sol. 1</td>
<td>1.67496</td>
<td></td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.40146</td>
<td>0.49547</td>
<td>4/0</td>
<td>221,295</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.43731</td>
<td>0.50620</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.49747</td>
<td>0.49329</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.49547</td>
<td>0.49747</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.50620</td>
<td>0.49329</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.50620</td>
<td>0.49747</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.49329</td>
<td>0.49547</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

of equations as given below.

Combustion application

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

The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 13. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 10. The comparison of $E_n$ obtained by the MLS and Evolutionary Approach is presented in Figure 11.

MLS is obtaining better results in terms of both running time and Euclidian norm. For this example, the values of $D_{MLS}$ and $D_{EA}$ are 0/7 and 7/0 respectively. This means
Table 13. Comparison of the results obtained by the MLS and evolutionary approach for combustion application

<table>
<thead>
<tr>
<th>Solution</th>
<th>MLS</th>
<th>Solution</th>
<th>Ev. Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_n$</td>
<td>$D_{MLS}$</td>
<td>Running time (mS)</td>
</tr>
<tr>
<td>Sol. 1</td>
<td>0.01506</td>
<td>0/7</td>
<td>860</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.01887</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.02539</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.02556</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.02870</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.03029</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.03052</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.03131</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 10. Convergence of the 8 solutions obtained by MLS for the combustion application

Figure 11. Comparison of euclidian norm for MLS and evolutionary approach for the combustion application

that no solution obtained by MLS is dominated by solutions obtained by the Evolutionary Approach while seven of the solutions obtained by the Evolutionary Approach are dominated by solutions obtained by MLS.
5.6. **Economics modelling application.** The following modelling problem is considered difficult and can be scaled up to arbitrary dimensions. The problem is given by the system of equations:

\[
\begin{align*}
(x_k + \sum_{i=1}^{n-k-1} x_i x_{i+k}) x_n - c_k &= 0, \quad 1 \leq k \leq n - 1 \\
\sum_{i=1}^{n-1} x_i + 1 &= 0
\end{align*}
\]

The constants \(c_k\) can be chosen randomly. We are considering the value 0 for the constants in our experiments.

We consider two instances \(e1\) and \(e2\) having 10 and 20 equations respectively.

**Table 14.** Comparison of results obtained by the MLS and evolutionary approach for the economics application \(e1\)

<table>
<thead>
<tr>
<th>Solution</th>
<th>(E_n)</th>
<th>(D_{\text{MLS}})</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>(E_n)</th>
<th>Ev. Approach</th>
<th>(D_{\text{EA}})</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.00294</td>
<td>0/0</td>
<td>266</td>
<td>Sol. 1</td>
<td>0.69997</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.00371</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>0.73992</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.00464</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>0.61761</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.00483</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>0.43294</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.00493</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>0.10015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.00716</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>0.02090</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.00772</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>0.11216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.00832</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>0.00376</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.6.1. **Economics modelling application \(e1\).** The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 14. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 12(a). The comparison of the \(E_n\) values obtained by the MLS and Evolutionary Approach is presented in Figure 13(a). As evident from the results presented, MLS is obtaining better results in terms of running time and Euclidian norm. For this example the values \(D_{\text{MLS}}\) and \(D_{\text{EA}}\) are having the same values: 0/0.

5.6.2. **Economics modelling application \(e2\).** The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 15. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 12(b). The comparison between \(E_n\) obtained by MLS and the Evolutionary Approach is presented in Figure 13(b). As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. For this example \(D_{\text{MLS}}\) and \(D_{\text{EA}}\) have the same value 0/0.

6. **Discussions and Conclusions.** The proposed Modified Line Search approach (MLS) seems to be very efficient for solving nonlinear equation systems. We first compared our approach for some simple equations systems having only two equations which were recently used for analyzing the performances of a new proposed method.

The results obtained using the proposed method were very promising outperforming some of the classical methods established in the literature (such as Newton, Broyden and secant methods).

The promising results obtained by the MLS approach for two-equation systems were the starting point and we extended the approach for high dimensional nonlinear equation systems. We also used some of the most well known applications such as: *interval arithmetic*
Figure 12. Convergence of the 8 solutions obtained by MLS for the economics modelling applications: (a)-e1 and (b)-e2

Figure 13. Comparison of euclidian norm for the MLS and evolutionary approach for the economics modelling applications: (a)-e1 and (b)-e2

benchmarks, neuropsychology, chemical equilibrium, kinematic, combustion and economic applications. All these applications consists of systems having a higher number of equations: 10 to 20 equations for the 5 different interval arithmetic benchmarks, 6 equations for the neuropsychology example, 5 equations for the chemical equilibrium application, 8
Table 15. Comparison of results obtained by the MLS and evolutionary approach for the economics modelling application e2

<table>
<thead>
<tr>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{MLS}$</th>
<th>Running time (mS)</th>
<th>Solution</th>
<th>$E_n$</th>
<th>$D_{EA}$</th>
<th>Running time (mS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sol. 1</td>
<td>0.00459</td>
<td>0/0</td>
<td>1,078</td>
<td>Sol. 1</td>
<td>0.41106</td>
<td>0/0</td>
<td>640,922</td>
</tr>
<tr>
<td>Sol. 2</td>
<td>0.00585</td>
<td></td>
<td></td>
<td>Sol. 2</td>
<td>0.21218</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 3</td>
<td>0.00844</td>
<td></td>
<td></td>
<td>Sol. 3</td>
<td>0.39753</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 4</td>
<td>0.00918</td>
<td></td>
<td></td>
<td>Sol. 4</td>
<td>0.22624</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 5</td>
<td>0.00917</td>
<td></td>
<td></td>
<td>Sol. 5</td>
<td>0.43207</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 6</td>
<td>0.01163</td>
<td></td>
<td></td>
<td>Sol. 6</td>
<td>0.20873</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 7</td>
<td>0.01072</td>
<td></td>
<td></td>
<td>Sol. 7</td>
<td>0.21682</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol. 8</td>
<td>0.01123</td>
<td></td>
<td></td>
<td>Sol. 8</td>
<td>0.17264</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

to 12 equations for the kinematic application, 10 equations for the combustion application and 10 to 20 equations for the economics application.

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

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

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

Acknowledgment. The first author acknowledges the support from the CNCSIS research grant IDEI 2412/2009.

REFERENCES


