International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1**, January **2012**

 ISSN 1349-4198
pp. 1–10-08062

SOLVING POLYNOMIAL SYSTEMS USING A MODIFIED LINE SEARCH APPROACH

Crina Grosan[1], Ajith Abraham[2,3] and Vaclav Snasel[2]

[1]Department of Computer Science
Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca 400084, Romania
cgrosan@cs.ubbcluj.ro

[2]Faculty of Electrical Engineering and Computer Science
VSB-Technical University of Ostrava
17. listopadu, 15/2172, Ostrava-Poruba 70833, Czech Republic
vaclav.snasel@vsb.cz

[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 interval 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. Several well known benchmarks are involved to show the performances of the proposed approach for solving polynomial systems. 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 $f(x) : \Re^n \to \Re$ 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 $\Omega = \prod_{i=1}^{n} [a_i, b_i] \subset \Re^n$.

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:

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

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 謀imensional 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:

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

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 = min_j + random * (max_j − min x_j);
    endfor
endfor
```

The *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:

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

where k refers to the iteration number.

The *Line_search()* technique may be written as 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 = random;
      $\alpha_k = 2 + \frac{3}{2^{k^2+1}}$
      $x_{ij}^{k+1} = x_{ij}^k + p_k \cdot \alpha_k$
    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:*

$$\textit{if } f(x_i^{k+1}) > f(x_i^k) \textit{ then } x_i^{k+1} = x_i^k$$

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 the iteration count starts again from 1 (this should not subsequently be related to the value of α 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 $t =$ *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 Ω be the search space. Consider n objective functions $f_1, f_2, \ldots, f_n$,

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

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

The multiobjective optimization problem is defined as:

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

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

Definition 4.1. *(Pareto dominance) Consider a maximization problem. Let x, y be two decision vectors (solutions) from Ω.*

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

(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:

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

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 nondominatd 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.

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 $P(t) \bigcup Q(t)$ and apply the reduction procedure if the allowed size of E is exceeded.

Step 2.4. Set $t = t + 1$;
$P(t) = Q(t)$

Step 3.

If 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 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 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

Benchmark		Number of variables	Range	References
1.	Interval i1	10	[–2, 2]	[21, 23, 28]
2.	Interval i2	20	[–1, 2]	[21, 23, 28]
3.	Interval i3	20	[–2, 2]	[21, 23, 28]
4.	Interval i4	10	[–1, 1]	[21, 23, 28]
5.	Interval i5	10	[–1, 1]	[21, 23, 28]
6.	Neurophysiology application	6	[–10, 10]	[45]
7.	Chemical equilibrium	5	[–10, 10]	[29]
8.	Kinematics kin1	12	[–10, 10]	[23]
9.	Kinematics kin2	8	[–10, 10]	[23]
10.	Combustion application	10	[–10, 10]	[29]
11.	Economics e1	10	[–10, 10]	[29]
12.	Economics e2	20	[–10, 10]	[29]

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

TABLE 2. MLS and evolutionary approach parameter settings

Parameter	**Value**									
MLS	*i1*	*i2, i3*	*i4, i5*	*neuro*	*chemical*	*kin1*	*kin2*	*combustion*	*e1*	*e2*
No of starting arbitrary points	100	100	100	100	100	100	100	100	100	100
No of re-starts (reinitializations)	10	10	10	10	10	20	10	10	10	10
No of iterations per each restarting phase	10	10	10	10	10	1	10	10	10	10
Evolutionary approach										
Population size	300	500	300	300	500	500	500	500	300	500
External set size	200	200	200	200	200	200	200	200	200	200
Number of generations	250	300	250	250	500	500	1000	300	300	300
Sigma (for mutation)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Tournament size	5	5	5	5	5	5	5	5	5	5

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:

$$\begin{cases} 0 = x_1 - 0.25428722 - 0.18324757x_4x_3x_9 \\ 0 = x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6 \\ 0 = x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} \\ 0 = x_4 - 0.19807914 - 0.15585316x_7x_1x_6 \\ 0 = x_5 - 0.44166728 - 0.19950920x_7x_6x_3 \\ 0 = x_6 - 0.14654113 - 0.18922793x_8x_5x_{10} \\ 0 = x_7 - 0.42937161 - 0.21180486x_2x_5x_8 \\ 0 = x_8 - 0.07056438 - 0.17081208x_1x_7x_6 \\ 0 = x_9 - 0.34504906 - 0.19612740x_{10}x_6x_8 \\ 0 = x_{10} - 0.42651102 - 0.21466544x_4x_8x_1 \end{cases}$$

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

TABLE 3. Comparison of results for the benchmark i1

Solution	MLS			Solution	Evolutionary Approach		
	En	D_{MLS}	Running time (mS)		*En*	D_{EA}	Running time (mS)
Sol. 1	0.19190	1/1	516	Sol. 1	0.74686	1/1	39,077
Sol. 2	0.21902			Sol. 2	0.68843		
Sol. 3	0.22084			Sol. 3	0.79977		
Sol. 4	0.19419			Sol. 4	0.80754		
Sol. 5	0.18906			Sol. 5	0.88129		
Sol. 6	0.21086			Sol. 6	0.84142		
Sol. 7	0.21813			Sol. 7	0.82571		
Sol. 8	0.22125			Sol. 8	0.81332		

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_{MLS} and D_{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).

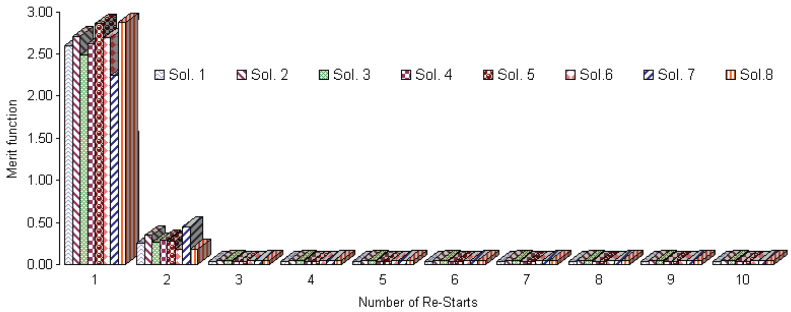
As evident from the results presented, MLS obtaines better results in terms of running time and Euclidian norm. D_{MLS} and D_{EA} have the same value: 0/0.

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

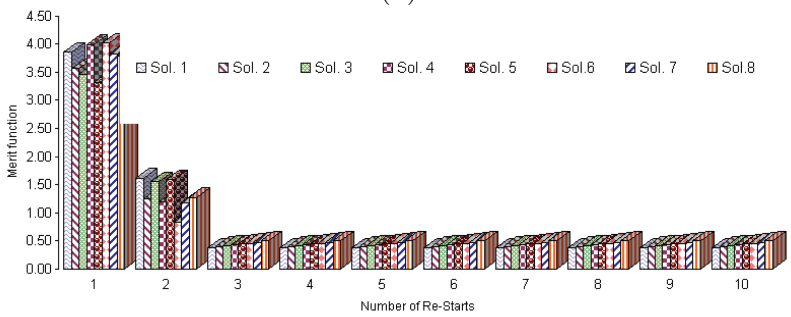
The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 5. The time displayed is in milliseconds.

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).

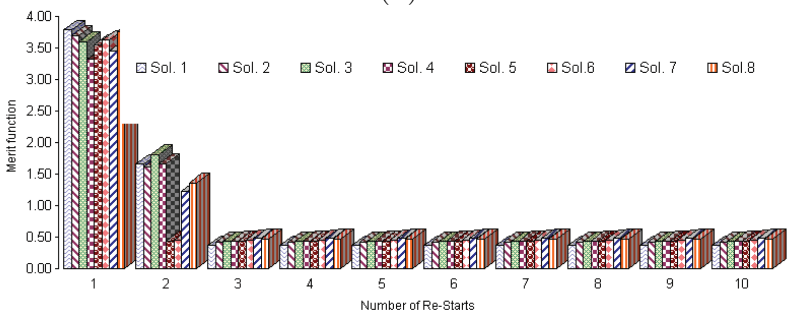
As evident from the results presented, MLS obtains better results in terms of running time and Euclidian norm. For this example, D_{MLS} and D_{EA} have the same value: 0/0.



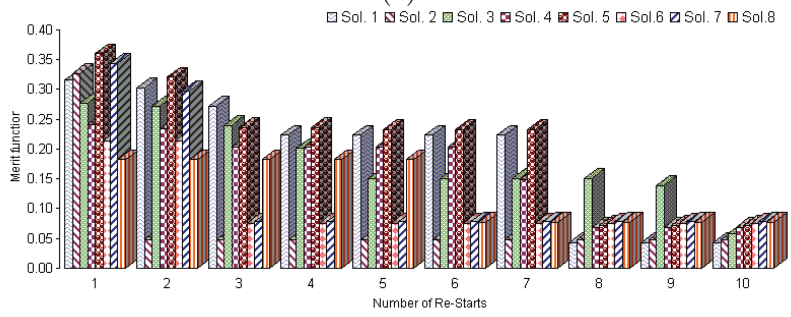
(a)



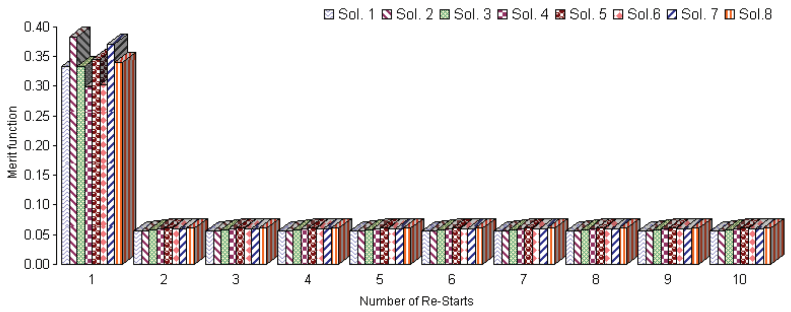
(b)



(c)



(d)



(e)

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

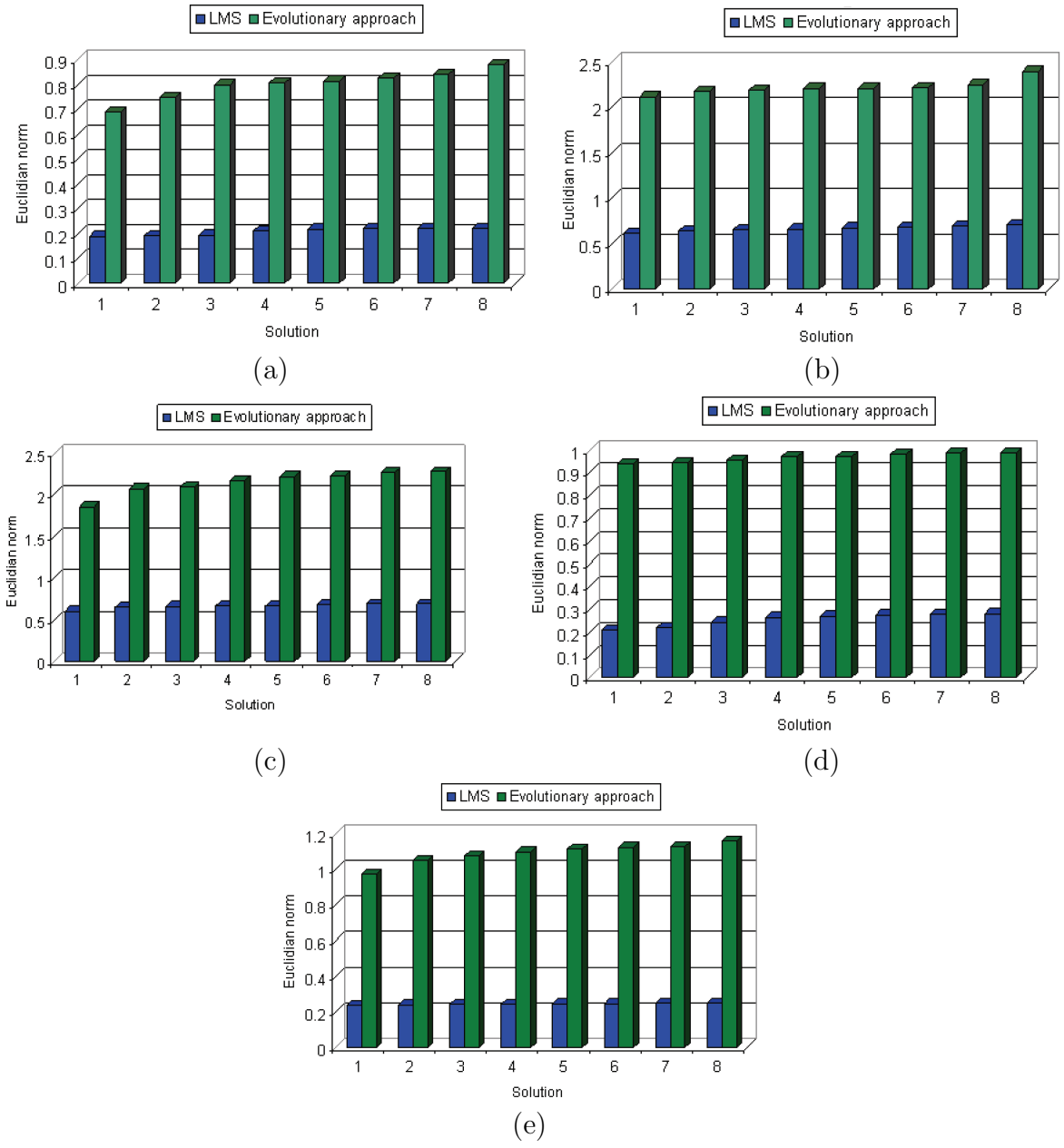


FIGURE 2. Comparison of euclidian norm for MLS and EA for interval benchmarks: (a)-benchmark i1, (b)-benchmark i2, (c)-benchmark i3, (d)-benchmark i4 and (e)-benchmark i5

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

$$\begin{cases} 0 = x_1^2 - 0.25428722 - 0.18324757x_4^2x_3^2x_9^2 \\ 0 = x_2^2 - 0.37842197 - 0.16275449x_1^2x_{10}^2x_6^2 \\ 0 = x_3^2 - 0.27162577 - 0.16955071x_1^2x_2^2x_{10}^2 \\ 0 = x_4^2 - 0.19807914 - 0.15585316x_7^2x_1^2x_6^2 \\ 0 = x_5^2 - 0.44166728 - 0.19950920x_7^2x_6^2x_3^2 \\ 0 = x_6^2 - 0.14654113 - 0.18922793x_8^2x_5^2x_{10}^2 \\ 0 = x_7^2 - 0.42937161 - 0.21180486x_2^2x_5^2x_8^2 \\ 0 = x_8^2 - 0.07056438 - 0.19612740x_1^2x_7^2x_6^2 \\ 0 = x_9^2 - 0.34504906 - 0.19612740x_{10}^2x_6^2x_8^2 \\ 0 = x_{10}^2 - 0.42651102 - 0.21466544x_4^2x_8^2x_1^2 \end{cases}$$

$$
\begin{cases}
0 = x_1 - 0.24863995 - 0.19594124 x_7 x_{10} x_{16} \\
0 = x_2 - 0.87528587 - 0.05612619 x_{18} x_8 x_{11} \\
0 = x_3 - 0.23939835 - 0.20177810 x_{10} x_7 x_{11} \\
0 = x_4 - 0.47620128 - 0.16497518 x_{12} x_{15} x_1 \\
0 = x_5 - 0.24711044 - 0.20198178 x_8 x_9 x_{16} \\
0 = x_6 - 0.33565227 - 0.15724045 x_{16} x_{18} x_{11} \\
0 = x_7 - 0.13128974 - 0.12384342 x_{12} x_{13} x_{15} \\
0 = x_8 - 0.45937304 - 0.18180253 x_{19} x_{15} x_{18} \\
0 = x_9 - 0.46896600 - 0.21241045 x_{13} x_2 x_{17} \\
0 = x_{10} - 0.57596835 - 0.16522613 x_{12} x_9 x_{13} \\
0 = x_{11} - 0.56896263 - 0.17221383 x_{16} x_{17} x_8 \\
0 = x_{12} - 0.70561396 - 0.23556251 x_{14} x_{11} x_4 \\
0 = x_{13} - 0.59642512 - 0.24475135 x_7 x_{16} x_{20} \\
0 = x_{14} - 0.46588640 - 0.21790395 x_{13} x_3 x_{10} \\
0 = x_{15} - 0.10607114 - 0.20920602 x_1 x_9 x_{10} \\
0 = x_{16} - 0.26516898 - 0.2137773 x_4 x_{19} x_9 \\
0 = x_{17} - 0.20436664 - 0.19838792 x_{20} x_{10} x_{13} \\
0 = x_{18} - 0.56003141 - 0.18114505 x_6 x_{13} x_8 \\
0 = x_{19} - 0.92894617 - 0.04417537 x_7 x_{13} x_{16} \\
0 = x_{20} - 0.57001682 - 0.17949149 x_1 x_3 x_{11}
\end{cases}
$$

FIGURE 3. Benchmark i2

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.60634	0/0	1,297	Sol. 1	2.22042	0/0	38,750
Sol. 2	0.65069			Sol. 2	2.25937		
Sol. 3	0.65608			Sol. 3	2.17129		
Sol. 4	0.66470			Sol. 4	2.27246		
Sol. 5	0.66435			Sol. 5	2.21490		
Sol. 6	0.67599			Sol. 6	1.85480		
Sol. 7	0.69210			Sol. 7	2.09113		
Sol. 8	0.68729			Sol. 8	2.06740		

TABLE 5. Comparison of the results obtained for the benchmark i3

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.61134	0/0	1,016	Sol. 1	2.20300	0/0	44,655
Sol. 2	0.63608			Sol. 2	2.20944		
Sol. 3	0.65165			Sol. 3	2.20199		
Sol. 4	0.65716			Sol. 4	2.17253		
Sol. 5	0.67047			Sol. 5	2.18607		
Sol. 6	0.67475			Sol. 6	2.24221		
Sol. 7	0.68942			Sol. 7	2.10679		
Sol. 8	0.70766			Sol. 8	2.38613		

TABLE 6. Comparison of the results obtained by MLS and the evolutionary approach for the benchmark i4

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		*En*	D_{EA}	Running time (mS)
Sol. 1	0.20734	0/6	953	Sol. 1	0.93678	6/0	366,404
Sol. 2	0.21872			Sol. 2	0.98004		
Sol. 3	0.24194			Sol. 3	0.98569		
Sol. 4	0.26271			Sol. 4	0.98555		
Sol. 5	0.26738			Sol. 5	0.96763		
Sol. 6	0.27376			Sol. 6	0.96981		
Sol. 7	0.27924			Sol. 7	0.94115		
Sol. 8	0.27768			Sol. 8	0.95304		

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.

$$\begin{cases} 0 = x_1^2 - 0.25428722 - 0.18324757x_4^3x_3^3x_9^3 + x_3^4x_9^7 \\ 0 = x_2^2 - 0.37842197 - 0.16275449x_1^3x_{10}^3x_6^3 + x_{10}^4x_6^7 \\ 0 = x_3^2 - 0.27162577 - 0.16955071x_1^3x_2^3x_{10}^3 + x_2^4x_{10}^7 \\ 0 = x_4^2 - 0.19807914 - 0.15585316x_7^3x_1^3x_6^3 + x_1^4x_6^7 \\ 0 = x_5^2 - 0.44166728 - 0.19950920x_7^3x_6^3x_3^3 + x_6^4x_3^7 \\ 0 = x_6^2 - 0.14654113 - 0.18922793x_8^3x_5^3x_{10}^3 + x_5^4x_{10}^7 \\ 0 = x_7^2 - 0.42937161 - 0.21180486x_2^3x_5^3x_8^3 + x_5^4x_8^7 \\ 0 = x_8^2 - 0.07056438 - 0.19612740x_1^3x_7^3x_6^3 + x_7^4x_6^7 \\ 0 = x_9^2 - 0.34504906 - 0.19612740x_{10}^3x_6^3x_8^3 + x_6^4x_8^7 \\ 0 = x_{10}^2 - 0.42651102 - 0.21466544x_4^3x_8^3x_1^3 + x_8^4x_1^7 \end{cases}$$

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		*En*	D_{EA}	Running time (mS)
Sol. 1	0.23610	0/1	1,000	Sol. 1	0.96990	1/0	120,546
Sol. 2	0.23820			Sol. 2	1.12068		
Sol. 3	0.24163			Sol. 3	1.04682		
Sol. 4	0.24384			Sol. 4	1.15779		
Sol. 5	0.24698			Sol. 5	1.11078		
Sol. 6	0.24618			Sol. 6	1.12241		
Sol. 7	0.24565			Sol. 7	1.07363		
Sol. 8	0.24702			Sol. 8	1.09669		

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 En 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:

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

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

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.01998	0/6	922	Sol. 1	0.34755	6/0	28,906
Sol. 2	0.05158			Sol. 2	0.38236		
Sol. 3	0.05608			Sol. 3	0.38443		
Sol. 4	0.05769			Sol. 4	0.36612		
Sol. 5	0.06200			Sol. 5	0.59597		
Sol. 6	0.06544			Sol. 6	0.34708		
Sol. 7	0.07451			Sol. 7	0.34823		
Sol. 8	0.08732			Sol. 8	0.20563		

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 En 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]):

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

where

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

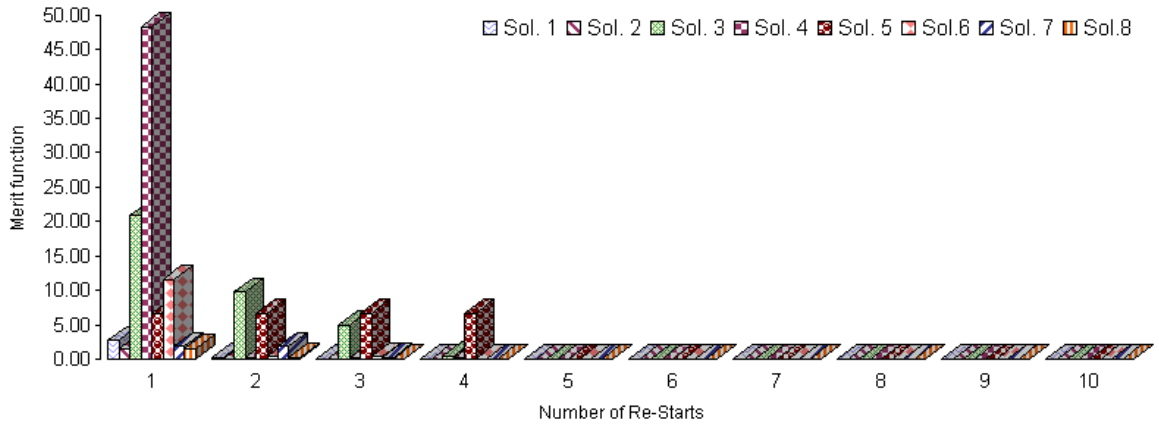


FIGURE 4. Convergence of the 8 solutions obtained by MLS for the neurophysiology application

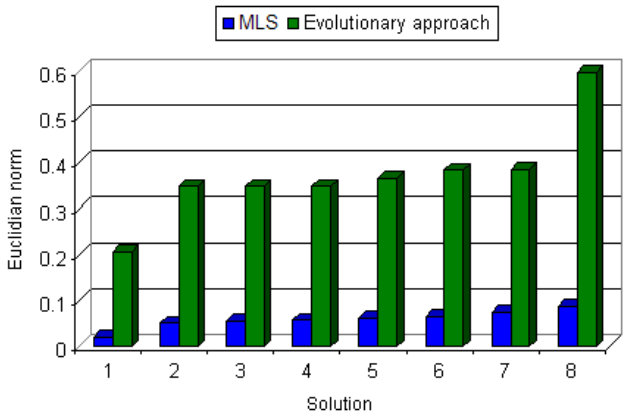


FIGURE 5. Comparison of euclidian norm obtained by MLS and the evolutionary approach for the neurophysiology application

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 Euclidian 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.

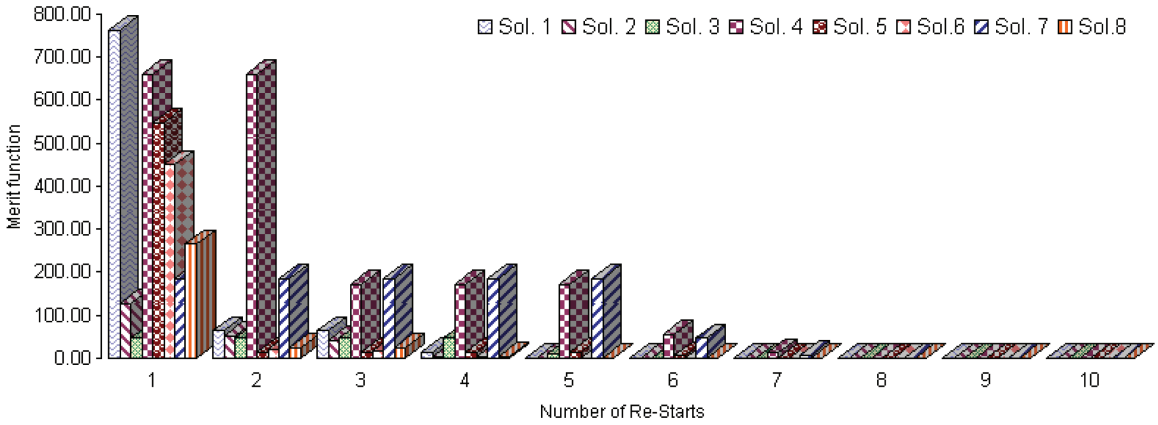


FIGURE 6. Convergence of the 8 solutions obtained by MLS for the chemical equilibrium application

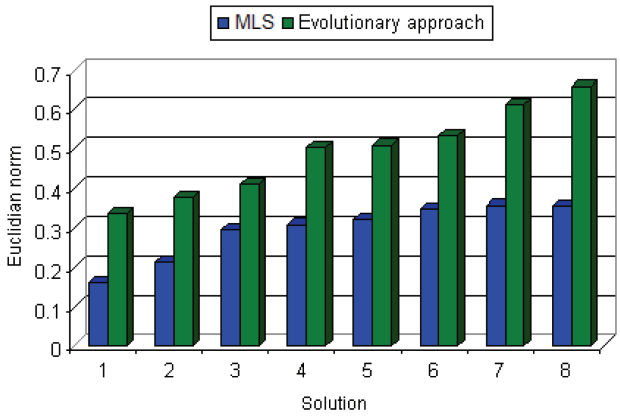


FIGURE 7. Comparison of euclidian norm for MLS and evolutionary approach for chemical equilibrium application

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.16072	0/4	922	Sol. 1	0.50300	4/0	28,906
Sol. 2	0.21222			Sol. 2	0.65838		
Sol. 3	0.29502			Sol. 3	0.61190		
Sol. 4	0.30787			Sol. 4	0.50826		
Sol. 5	0.32114			Sol. 5	0.33594		
Sol. 6	0.34612			Sol. 6	0.40933		
Sol. 7	0.35578			Sol. 7	0.37625		
Sol. 8	0.35341			Sol. 8	0.53241		

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.

$$\text{kin1 application} \begin{cases} s_2c_5s_6 - s_3c_5s_6 - s_4c_5s_6 + c_2c_6 + c_3c_6 + c_4c_6 = 0.4077 \\ c_1c_2s_5 + c_1c_3s_5 + c_1c_4s_5 + s_1c_5 = 1.9115 \\ s_2s_5 + s_3s_5 + s_4s_5 = 1.9791 \\ c_1c_2 + c_1c_3 + c_1c_4 + c_1c_2 + c_1c_3 + c_1c_2 = 4.0616 \\ s_1c_2 + s_1c_3 + s_1c_4 + s_1c_2 + s_1c_3 + s_1c_2 = 1.7172 \\ s_2 + s_3 + s_4 + s_2 + s_3 + s_2 = 3.9701 \\ s_i^2 + c_i^2 = 1, \quad 1 \leq i \leq 6 \end{cases}$$

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 10. Results obtained by the MLS and evolutionary approach for kinematics application kin1

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.64788	0/8	937	Sol. 1	6.96304	8/0	109,327
Sol. 2	0.75080			Sol. 2	6.72439		
Sol. 3	0.74959			Sol. 3	6.38505		
Sol. 4	0.83451			Sol. 4	6.79585		
Sol. 5	0.83541			Sol. 5	6.40160		
Sol. 6	0.83830			Sol. 6	6.88722		
Sol. 7	0.84435			Sol. 7	6.44937		
Sol. 8	0.85331			Sol. 8	6.35426		

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:

$$\text{kin2 application} \begin{cases} x_i^2 + x_{i+1}^2 - 1 = 0 \\ a_{1i}x_1x_3 + a_{2i}x_1x_4 + a_{3i}x_2x_3 + a_{4i}x_2x_4 \\ +a_{5i}x_2x_7 + a_{6i}x_5x_8 + a_{7i}x_6x_7 + a_{8i}x_6x_8 \\ +a_{9i}x_1 + a_{10i}x_2 + a_{11i}x_3 + a_{12i}x_4 + a_{13i}x_5 + a_{14i}x_6 \\ +a_{15i}x_7 + a_{16i}x_8 + a_{17i} = 0 \\ 1 \leq i \leq 4 \end{cases}$$

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

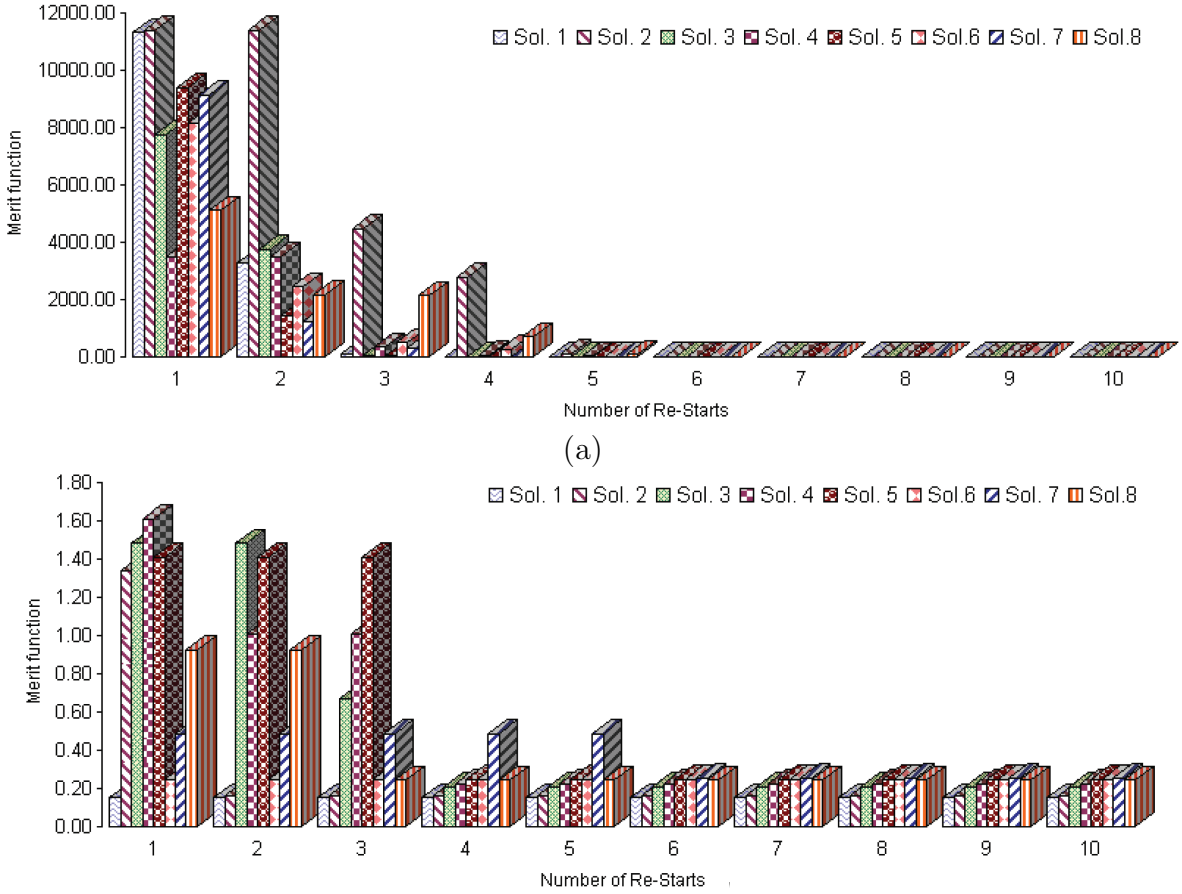


FIGURE 8. Convergence of the 8 solutions obtained by MLS for the kinematics applications: (a)-kin1 and (b)-kin2

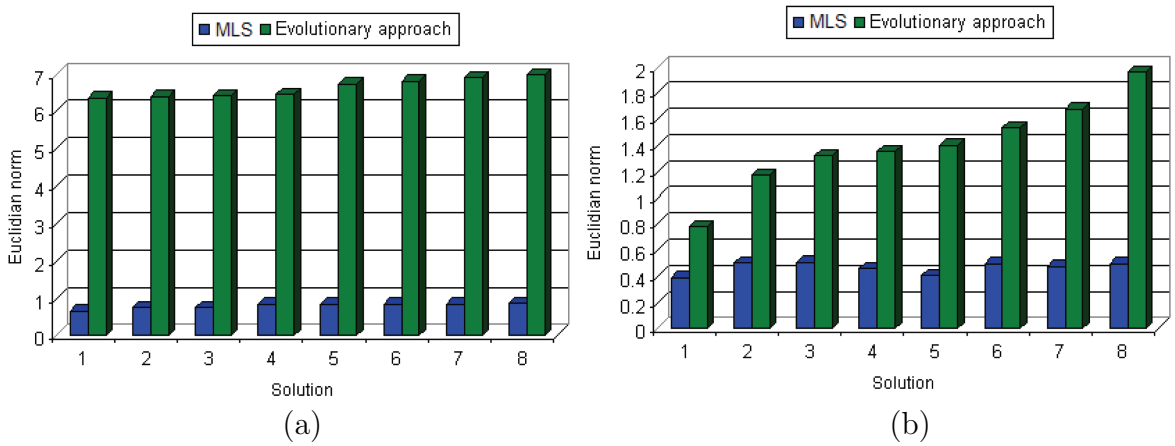


FIGURE 9. Comparison of euclidian norm for MLS and evolutionary approach for kinematics applications: (a)-kin1 and (b)-kin2

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_{ki} for the kinematic example kin2

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

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.38881	0/4	969	Sol. 1	1.67496	4/0	221,295
Sol. 2	0.40146			Sol. 2	1.34899		
Sol. 3	0.45731			Sol. 3	1.31849		
Sol. 4	0.47221			Sol. 4	1.53427		
Sol. 5	0.49747			Sol. 5	1.95724		
Sol. 6	0.49547			Sol. 6	1.39915		
Sol. 7	0.50620			Sol. 7	0.77671		
Sol. 8	0.49329			Sol. 8	1.16787		

of equations as given below.

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

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 En 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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		En	D_{EA}	Running time (mS)
Sol. 1	0.01506	0/7	860	Sol. 1	0.19273	7/0	151,123
Sol. 2	0.01887			Sol. 2	0.28188		
Sol. 3	0.02539			Sol. 3	0.24526		
Sol. 4	0.02556			Sol. 4	0.32531		
Sol. 5	0.02870			Sol. 5	0.27724		
Sol. 6	0.03029			Sol. 6	0.06548		
Sol. 7	0.03052			Sol. 7	0.11600		
Sol. 8	0.03131			Sol. 8	0.15219		

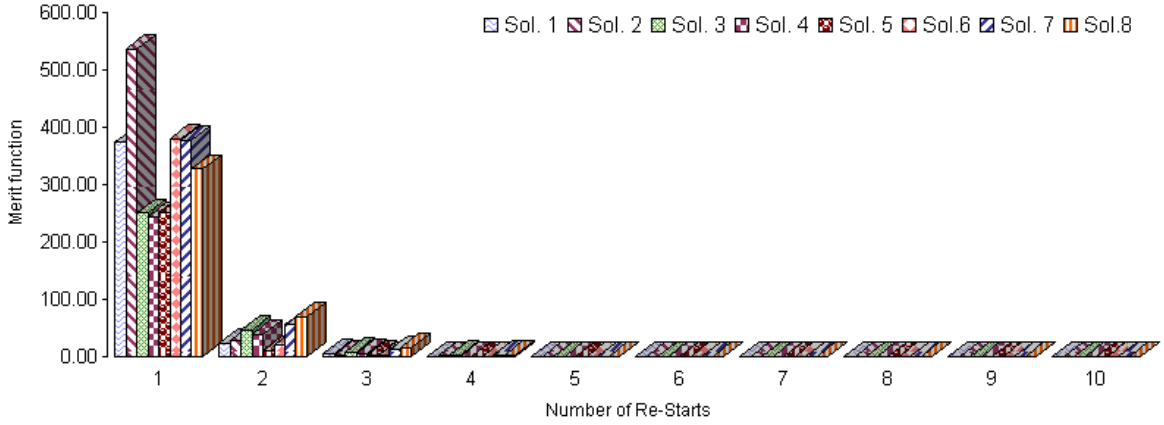


FIGURE 10. Convergence of the 8 solutions obtained by MLS for the combustion application

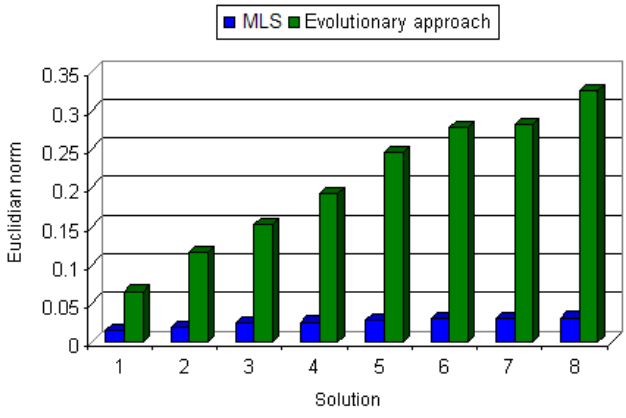


FIGURE 11. Comparison of euclidian norm for MLS and evolutionary approach for the combustion application

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:

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

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		*En*	D_{EA}	Running time (mS)
Sol. 1	0.00294	0/0	266	Sol. 1	0.69997	0/0	396,399
Sol. 2	0.00371			Sol. 2	0.73992		
Sol. 3	0.00464			Sol. 3	0.61761		
Sol. 4	0.00483			Sol. 4	0.43294		
Sol. 5	0.00493			Sol. 5	0.10015		
Sol. 6	0.00716			Sol. 6	0.02090		
Sol. 7	0.00772			Sol. 7	0.11216		
Sol. 8	0.00832			Sol. 8	0.60376		

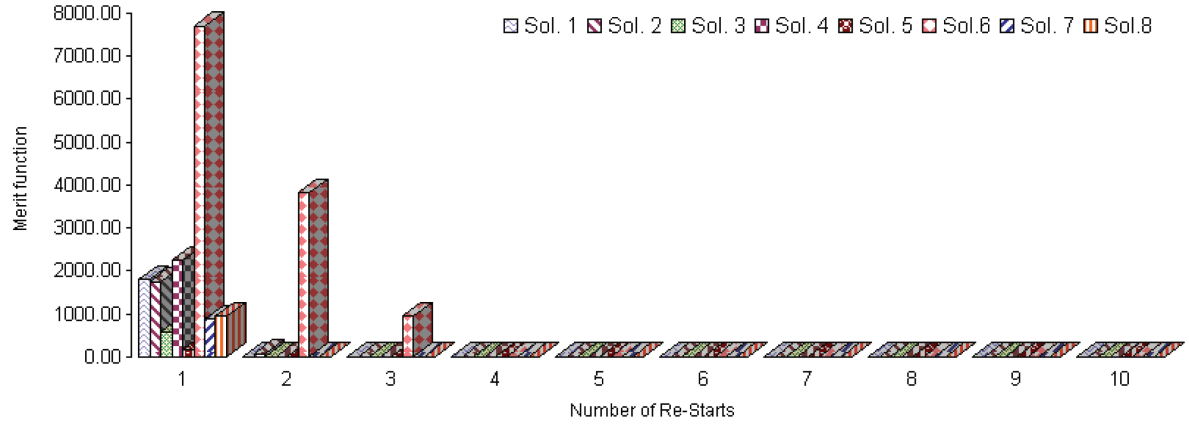
5.6.1. *Economics modelling application e1.* The comparison of the results obtained by the MLS and Evolutionary Approach are presented in Table 14. The time displayed is in milliseconds. The convergence of all 8 solutions obtained by MLS is depicted in Figure 12(a). The comparison of the *En* values 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_{MLS} and D_{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 *En* 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_{MLS} and D_{EA} have the same value 0/0.

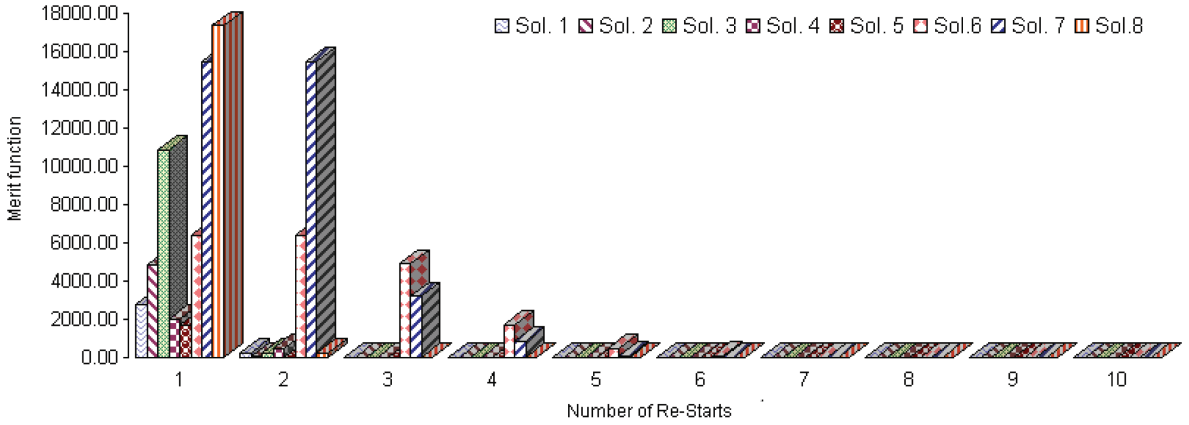
6. **Discussions and Conclusions.** The proposed Modified Line Search approach (MLS) seems to be very efficient for solving nonlinear equation systems. We first compared our approach for some simple equations systems having only two equations which were recently used for analyzing the performances of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

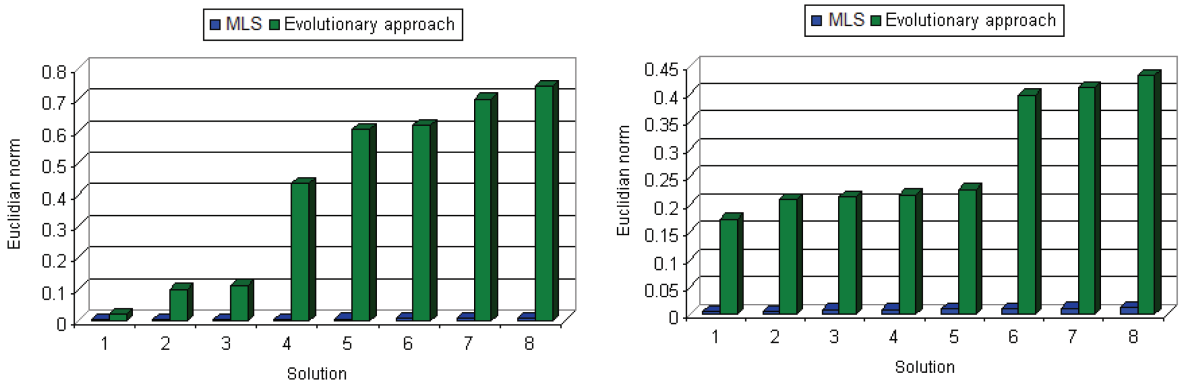


(a)



(b)

FIGURE 12. Convergence of the 8 solutions obtained by MLS for the economics modelling applications: (a)-e1 and (b)-e2



(a)

(b)

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

Solution	MLS			Solution	Ev. Approach		
	En	D_{MLS}	Running time (mS)		*En*	D_{EA}	Running time (mS)
Sol. 1	0.00459	0/0	1,078	Sol. 1	0.41106	0/0	640,922
Sol. 2	0.00585			Sol. 2	0.21218		
Sol. 3	0.00844			Sol. 3	0.39753		
Sol. 4	0.00918			Sol. 4	0.22624		
Sol. 5	0.00917			Sol. 5	0.43207		
Sol. 6	0.01163			Sol. 6	0.20873		
Sol. 7	0.01072			Sol. 7	0.21682		
Sol. 8	0.01123			Sol. 8	0.17264		

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 inequations systems and system of differential equations, which are part of our future research work.

REFERENCES

[1] B. W. Bader, Tensor-Krylov methods for solving large-scale systems of nonlinear equations, *SIAM Journal of Numerical Analysis*, vol.43, no.3, pp.1321-1347, 2005.

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

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

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

[5] J. E. Denis, On Newton-like method and nonlinear simultaneous replacements, *SIAM Journal of Numerical Analysis*, vol.4, pp.103-108, 1967.

[6] J. E. Denis, On Newton-like methods, *Numerical Mathematics*, vol.11, pp.324-330, 1968.

[7] J. E. Denis, On the convergence of Broydens method for nonlinear systems of equations, *Mathematics of Computation*, vol.25, pp.559-567, 1971.

[8] J. E. Denis and H. Wolkowicz, Least-change secant methods, sizing, and shifting, *SIAM Journal of Numerical Analysis*, vol.30, pp.1291-1314, 1993.

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

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

[11] L. J. Eshelman, R. A. Caruna and J. D. Schaffer, Biases in the crossover landscape, *Proc. of the 3rd International Conference on Genetic Algorithms*, 1989.

[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA, USA, 1989.
[13] T. N. Grapsa and M. N. Vrahatis, The implicit function theorem for solving systems of nonlinear equations in R2, *International Journal of Computer Mathematics*, vol.28, pp.171-181, 1989.
[14] T. N. Grapsa and M. N. Vrahatis, A dimension – Reducing method for solving systems of nonlinear equations in R^n, *International Journal of Computer Mathematics*, vol.32, pp.205-216, 1990.
[15] T. N. Grapsa, M. N. Vrahatis and T. C. Bountis, Solving systems of nonlinear equations in R^n using a rotating hyperplane in R^{n+1}, *International Journal of Computer Mathematics*, vol.35, pp.133-151, 1990.
[16] T. N. Grapsa and M. N. Vrahatis, A new dimension – Reducing method for solving systems of nonlinear equations, *International Journal of Computer Mathematics*, vol.55, pp.235-244, 1995.
[17] T. N. Grapsa, M. N. Vrahatis, T. N. Grapsa and M. N. Vrahatis, A dimension – Reducing method for unconstrained optimization, *Journal of Computational and Applied Mathematics*, vol.66, pp.239-253, 1996.
[18] T. N. Grapsa and M. N. Vrahatis, Dimension reducing methods for systems of nonlinear equations and unconstrained optimization: A review, *Recent Advances in Mechanics and Related Fields*, pp.215-225, 2003.
[19] C. Grosan and A. Abraham, A new approach for solving nonlinear equations systems, *IEEE Transactions on Systems Man and Cybernetics – Part A*, vol.38, no.3, pp.698-714, 2008.
[20] C. Grosan and A. Abraham, Multiple solutions for a system of nonlinear equations, *International Journal of Innovative Computing, Information and Control*, vol.4, no.9, pp.2161-2170, 2008.
[21] P. Van Hentenryck, D. McAllester and D. Kapur, Solving polinomial systems using a branch and prune approach, *SIAM Journal of Numerical Analysis*, vol.34, no.2, pp.797-827, 1997.
[22] M. J. Hirsch, C. N. Meneses, P. M. Pardalos and M. G. C. Resende, Global optimization by continuous grasp, *Optimization Letters*, vol.1, no.2, pp.201-212, 2007.
[23] H. Hong and V. Stahl, Safe starting regions by fixed points and tightening, *Computing*, vol.53, pp.323-335, 1994.
[24] D. Manocha, Solving systems of polynomial equations, *IEEE Computer Graphics and Applications*, vol.16, pp.46-55, 1994.
[25] J. M. Martinez, Algorithms for solving nonlinear systems of equations, *Continuous Optimization: The State of the Art*, pp.81-108, 1994.
[26] A. A. Martynyuk, An exploration of polydynamics of nonlinear equations on time scales, *ICIC Express Letters*, vol.2, no.2, pp.155-160, 2008.
[27] K. Meintjes and A. P. Morgan, Chemical equilibrium systems as numerical test problems, *ACM Transaction on Mathematical Software*, vol.16, pp.143-151, 1990.
[28] R. E. Moore, *Methods and Applications of Interval Analysis*, SIA, Philadelphia, 1979.
[29] A. P. Morgan, Computing all solutions to polynomial systems using homotopy continuation, *Applied Mathematics an Computation*, vol.24, pp.115-138, 1987.
[30] A. P. Morgan, *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.
[31] A. P. Morgan, Polynomial continuation and its relationship to the symbolic reduction of polynomial systems, in *Symbolic and Numerical Computation for Artificial Intelligence*, B. Donald et al. (eds.), New York, Academic Press, 1992.
[32] P. Y. Nie, A null space method for solving system of equations, *Applied Mathematics and Computation*, vol.149, no.1, pp.215-226, 2004.
[33] P. Y. Nie, An SQP approach with line search for a system of nonlinear equations, *Mathematical and Computer Modelling*, vol.43, pp.368-373, 2006.
[34] J. Nielson and B. Roth, On the kinematic analysis of robotic mechanisms, *The International Journal of Robotics Research*, vol.18, no.12, pp.1147-1160, 1999.
[35] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.
[36] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 2002.
[37] W. C. Rheinboldt, Methods for solving systems of equations, *Reg. Conf. Ser. in Appl. Math*, vol.14, 1974.
[38] B. Salimbahrami and B. Lohmann, Order reduction of large scale second-order systems using Krylov subspace methods, *Linear Algebra and Its Applications*, vol.415, pp.385-905, 2006.

[39] J. D. Schaffer and A. Morishima, An adaptive crossover distribution mechanism for genetic algorithms, *Proc. of the 2nd International Conference on Genetic Algorithms*, pp.36-40, 1987.
[40] T. W. Sederberg and T. Nishita, Curve intersection using Bezier clipping, *Computer-Aided Design*, vol.22, no.9, pp.538-549, 1990.
[41] A. J. Sommese and C. W. Wampler, *The Numerical Solution of Systems of Polynomials: Arising in Engineering and Science*, World Scientific, 2005.
[42] W. M. Spears and K. A. De Jong, On the virtues of uniform crossover, *Proc. of the 4th International Conference on Genetic Algorithms*, pp.230-236, 1991.
[43] G. Syswerda, Uniform crossover in genetic algorithms, *Proc. of the 3rd Conference on Genetic Algorithms*, 1989.
[44] R. E. Steuer, Multiple criteria optimization; theory, computation, and application, in *Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics*, A. J. Osiadacz (ed.), New York, John Wiley and Sons, 1986.
[45] J. Verschelde, P. Verlinden and R. Cools, Homotopies exploiting Newton polytopes for solving sparse polynomial systems, *SIAM Journal of Numerical Analysis*, vol.31, no.3, pp.915-930, 1994.
[46] X. Zhong, T. Zhang and Y. Shi, Oscillation and nonoscillation of neutral difference equation with positive and negative coefficients, *International Journal of Innovative Computing, Information and Control*, vol.5, no.5, pp.1329-1342, 2009.
[47] G. Zhou and P. Zhu, Iterative algorithms for solving repeated root of nonlinear equation by modified Adomian decomposition method, *ICIC Express Letters*, vol.4, no.2, pp.595-600, 2010.
[48] J. H. Wilkinson, The evaluation of the zeros of Ill-conditioned polynomials, Parts I and II, *Numerical Mathematics*, vol.1, pp.150-166; pp.167-180, 1959.