



Predictive Memetic Algorithm (PMA) for Combinatorial Optimization in Dynamic Environments

Stephen M. Akandwanaho and Serestina Viriri^(✉)

School of Mathematics, Statistics and Computer Science,
University of KwaZulu-Natal, Durban, South Africa
{akandwanahos,viriris}@ukzn.ac.za

Abstract. A prediction mechanism for Memetic Algorithm is presented in this paper. The Predictive Memetic Algorithm (PMA) uses a nonlinear regression method to estimate the parameters used by the algorithm to obtain good solutions in a dynamic and stochastic environment. The algorithm is applied to nonlinear data sets and performance is compared with genetic and simulated annealing algorithms. When compared with the existing methods, the proposed method generates a relatively small error difference after prediction thereby proving its superior performance. A dynamic stochastic environment is used for experimentation, so as to determine the efficacy of the algorithm on non-stationary problem environments.

Keywords: Memetic Algorithm · Local search · Optimization
Non-linear regression · Prediction

1 Introduction

Most of the artificial intelligence mechanisms draw inspiration from nature. These mechanisms follow the natural evolutionary process for their internal transformations. For example, Genetic algorithm (GA) which is based on the concept of natural genes and evolutionary inheritance process has been applied to solve many problems [1]. However, due to its inherent inefficiencies such as premature convergence [2], Memetic algorithm (MA) was developed address them.

Although MA builds on genetic evolution, it presents another form of evolution called memetic evolution [3–5]. This is where memes are transmitted from brain to brain through interaction and experience. A meme represents a unit of information in MA that can be passed on from one individual to another, such as ideas, beliefs, attitudes and principles. These can then be multiplied across generations.

Moscato [1] gives some of the benefits of Memetic Computing, and the efficacy of hybrid genetic algorithms in generation high quality solutions for complex and dynamic problems. One of their major benefits is the ability to refine solutions

by performing a neighborhood search to avert premature convergence in global search methods. MA leverages both global and local search methods to improve the search, thereby offsetting the weaknesses of individual methods. For example, the global method does not perform search intensification to exploit the search space. By the same token, the local methods do not perform diversification to explore the search. These limitations are alleviated in MA.

MA has been extensively applied to solve a variety of problems in optimization [6–12]. However, MAs have been majorly applied to stationary environments, and yet the current problems have become more complex, continuous and dynamic [13–15], hence an increasing demand for more intelligent mechanisms.

The predictive approaches in evolutionary algorithms help to track moving optima in stochastic problem environments. Therefore, in this paper prediction is introduced in MA for dynamic problem environments, based on nonlinear regression method [18, 19], which lends itself to rugged landscapes and shifting optima in capricious search.

The remainder of this article is organized as follows. The Sect. 2 introduces the problem, Sect. 3 presents a review of related work. The Sect. 5 introduces the proposed architecture. Results are discussed in the Sect. 6. The Sect. 7 covers the conclusion, limitations and potential future research.

2 Problem Definition

In a stochastic environment, it is difficult to predict the optimal outcome, especially when the optima is constantly moving within the search space. This increases the complexity and intractability of the problem. In combinatorial optimization, multiple solutions are generated for an optimization problem. The problem can be defined by the objective function with a feasible region in which alternative solutions are found. In this context, the problem at hand is nonlinear, which depicts its stochasticity, and the optimization objective is to find a solution with the smallest error, denoted by ε . R^p is the parameter space, R^m is the solution space and $f(x, \theta)$ is a nonlinear function of θ . Therefore, the problem is to estimate θ in a standard nonlinear regression model in Eqs. 1 and 2. σ and N are nonlinear parameter and population sample respectively.

$$y = f(x, \theta) + \varepsilon \quad (1)$$

$$\varepsilon \sim N(0, \sigma^2) \quad (2)$$

For $x \in R^m$, the function of x and θ denote nonlinear where $\theta \in R^p$ [29]. Consider a solution set, where y_i, x_i and $i = 1, 2, 3, \dots, N$. We then minimize θ as follows:

$$T(\hat{\theta}) \leq T(\theta) \quad (3)$$

$$T(\theta) = \sum_{i=1}^n [y_i - f(x_i, \theta)]^2 \quad (4)$$

$$\text{Min}T(\theta) \tag{5}$$

The nonlinear estimation problem is for intents and purposes an optimization problem, hence this requires the minimization of the objective function, $T(\theta)$, in Eqs. 3, 4 and 5 for solutions in the sample.

3 Related Work

Zhou et al. [20] presented an adaptive hill climbing strategy implemented with MA for dynamic optimization problems. The crossover-based hill climbing and mutation-based hill climbing are combined to increase search diversity, although there is high computational complexity arising from other integrated methods, such as adaptive dual mapping and triggered random. [12, 14–16] introduced other techniques in prediction to foretell Pareto front's presence in problems of multiobjective nature. Pareto's approach draws on past information to read patterns and make predictions, in addition to using the apriori technique. This is similar to the multiobjective method presented in [17–20], where the population is reused. The method draws on past details to reach good solutions. The past record of an individual is characterized by the changes that have taken place and the perturbation record helps to determine the behavior of subsequent solutions. However, different variations are factored in, over and above, the population parameters and time.

The algorithm presented in [21, 22] estimates how the changes will be at a certain period of time. The function of Osmera and Knapsack problems were used to test the approach. The optimum in the last generation is used to shape the optimum in the next generation. This method uses two predictors, namely noisy predictor and perfect predictor. The predictors use observations and data of individuals. In a similar vein, Weicker [23] shows how the decision process at every stage provides leverage when solving recurrent problems. This method draws patterns through deeper studying of relationships between the problem and possible solutions in the sample. Kalman algorithm presented by Stroud [24] gives the importance of evaluating the strength of individuals through fitness values and using this analysis for better decision making. Every information is useful from the specific elements in the search to the search space. However, in nonlinear situations, the current information in linear samples may not be sufficient to accurately keep track of moving optima.

Karaman et al. [25] presented an online dynamic optimization method. Time is one of the factors used to track changes and draw patterns. The method was applied to problems, such as the vehicle routing and scheduling. In a similar vein, Bosman [26, 27] uses machine learning, evolutionary methods and statistics to forecast environment behaviors. In order to prevent premature convergence, intensification and diversification are used to enhance the accuracy of the prediction outcome. The majority of related work in the literature, shows that a lot of predictive research is directed towards evolutionary algorithms, such as genetic algorithms and others but few on MAs.

4 Dynamic Landscapes

Dynamic environments are explained extensively in [22–24, 28], especially on how the environments change, why they change and how they change. Although most of the research has been heavy in stationary problems, the pendulum of research is increasingly shifting to dynamic problems due a proliferation of such problems. Hongfeng et al. [41] presented an adaptive hill climbing method, where greedy crossover and steepest mutation are combined to produce good solutions. Dynamic problems can be either exogenous or endogenous. Weicker [42] defines exogenous as those dynamics that are directly inherent to the problem, while endogenous dynamics come from the evolutionary process. The population keeps on evolving, as individuals shift positions in the dynamic nature of the environment.

Figure 1 clearly shows that the shifting optima is always on the move. Intelligent methods would be able to track these dynamic alleles. The drifting landscapes are compounded by the morphology changes of individuals in the population.

5 Proposed Predictive MA

Algorithm 1 shows a proposed prediction model implemented in Predictive MA. The objective of PMA is to find θ . In case of any signs of environmental change, PMA finds θ that suits the available dataset, therefore, using apriori, θ is approximated. After these changes have occurred, as seen in Fig. 1, ε is defined. The error defines the difference between the estimated and non estimated. However, in error prediction, the difference between the generational change and estimation is registered to provide parameters for dispensing with weak solutions. The smallest error remains the aim of the function, and if the smallest error is recorded, it is bench-marked for the next generation. High quality approximation of parameters is important to achieve accuracy, as presented by Simoes and Costa [43].

The proposed model is intrinsically nonlinear. The parameters pertaining to the vector are estimated, and the minimization of error is important for increasing accuracy and performance. MA follows the evolutionary process using the

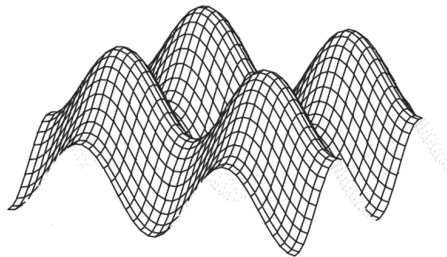


Fig. 1. The nature of candidate solutions in a dynamic setting [42].

```

Population initialization;
Create empty memory;
Evaluate population;
Predict operators;
for each individual,  $i$  in  $P$  do
    Apply operators;
    Apply LocalSearch;
    if  $i > j$  then
        Store  $j$ ;
        if  $i < j$  then
            Store  $i$ ;
        end
    end
    if change in landscape then
        for each individual,  $i$  in  $P$  do
            Predict operators;
            Apply operators;
            Apply LocalSearch if  $i > j$  then
                Store  $j$ ;
                if  $i < j$  then
                    Store  $i$ ;
                end
            end
        end
    end
end
Update memory;
Select best,  $i$ ;
Define next population;
Stop;

```

Algorithm 1. Proposed Memetic Algorithm

parameters as shown in Algorithm 1. Nonlinear regression applies to both linear and nonlinear problems. Many problems are inherently nonlinear which makes it hard for linear functions to produce good results on nonlinear behavioral patterns. The choice of nonlinear is also based on robust data exploitation capacity, manipulating data and behaviors to make accurate inferences. The proposed model approximates the parameters of θ and makes predictions.

6 Experimental Results

The standard problems of nonlinear regression [44–46] were used in this study to conduct the experiments. Results were obtained from 100 iterations, and in Tables 1, 2 and 3, PMA which is the proposed method, performs better by yielding the lowest value of θ , hence achieving the minimization of the objective function.

Table 1. θ is solved by different methods GA, SA and PMA

PMA yields better results			
Iteration	GA	SA	PMA
1	7.098670	9.448511	6.815231
2	7.098670	9.448511	6.815231
3	7.098670	9.448511	6.815231
4	7.098670	9.448511	6.815231
5	7.098670	9.448511	6.815231
6	7.098670	9.448511	6.815231
7	7.098670	9.448511	6.815231
8	7.098670	9.448511	6.815231
9	7.098670	9.448511	6.815231
10	7.801110	9.448511	7.640785
11	7.851110	9.448511	7.640785
12	7.762110	9.448511	7.640785
13	7.873110	8.275364	7.640785
14	7.721110	10.217881	7.640785
15	7.881110	10.217881	7.640785
16	7.901110	10.217881	7.640785
17	7.791110	10.393921	7.640785
18	7.881110	10.393921	7.640785
19	7.698110	12.080187	7.640785
20	7.771110	12.080187	7.640785
21	7.891110	11.524730	7.640785
22	7.721110	11.524730	7.640785
23	7.431110	11.524730	7.640785
24	7.791110	10.448512	7.640785
25	7.631110	10.448512	7.523401
26	7.721110	10.448512	7.523401
27	7.651110	9.217811	7.523401
28	7.701110	9.217811	7.523401
29	7.681110	9.217811	7.523401
30	7.811110	9.217811	7.523401
31	7.791110	9.217811	7.523401
32	7.601110	9.217811	7.523401
33	7.901110	9.217811	7.523401
34	7.781110	9.393920	7.523401
35	7.601110	9.407372	7.523401
36	7.613210	9.407372	7.523401
37	7.703110	9.866600	7.523401
38	7.911110	9.866600	7.523401
39	7.601110	9.866600	7.523401
40	7.723110	9.866600	7.523401
41	7.601110	9.866600	7.523401

Table 2. θ is solved by different methods GA, SA and PMA

PMA yields better results			
Iteration	GA	SA	PMA
42	7.10110	9.36222	6.421933
43	7.10110	9.36222	6.421933
44	7.10110	9.36222	6.421933
45	7.10110	9.36222	6.421933
46	7.10110	9.36222	6.421933
47	7.10122	9.36222	6.421933
48	7.10122	9.36222	6.421933
49	7.10122	9.36222	6.421933
50	7.10122	9.84968	6.421933
51	7.10122	8.89983	6.421933
52	7.10122	10.87476	6.421933
53	7.10122	10.28331	6.421933
54	7.10122	10.28331	6.421933
55	7.10122	9.82237	6.421933
56	7.10122	9.82237	6.421933
57	7.10122	9.82237	6.421933
58	7.10122	9.82237	6.421933
59	7.10122	9.82237	6.421933
60	7.11055	8.22502	6.421933
61	7.11055	8.22502	6.421933
62	7.11055	8.22502	6.421933
63	7.11055	8.22502	6.421933
64	7.11055	8.22502	6.421933
65	7.11055	8.22502	6.421933
66	7.11055	8.22502	6.421933
67	7.11055	8.22502	6.421933
68	7.11055	8.22502	6.421933
69	7.11055	8.22502	6.421933
70	7.11055	8.22502	6.421933
71	7.11055	9.22502	6.421933
72	7.11055	9.22502	6.542042
73	7.12066	9.22502	6.542042
74	7.12066	9.30006	6.542042
75	7.12066	9.30006	6.542042
76	7.12066	9.30006	6.542042
77	7.12066	9.30006	6.542042
78	7.12066	9.30006	6.542042
79	7.12066	9.30006	6.542042
80	7.470024	9.30006	6.542042

Table 3. θ is solved by different methods GA, SA and PMA

PMA yields better results			
Iteration	GA	SA	PMA
81	6.871684	8.30006	6.771297
82	7.005943	8.30006	6.771297
83	7.005943	8.5730	6.771297
84	7.005943	8.3479	6.771297
85	7.005943	8.62329	6.771297
86	7.005943	9.62329	6.771297
87	7.140203	9.62329	6.771297
88	8.518115	9.02486	7.621498
89	8.518115	8.02486	7.621498
90	8.518115	8.21780	7.621498
91	8.518115	9.21780	7.621498
92	8.28100	9.21780	7.621498
93	9.28100	9.39392	7.621498
94	9.28100	9.39392	7.621498
95	9.28100	10.08018	7.621498
96	9.66239	10.08018	9.11152
97	9.66239	10.52473	10.41042
98	9.66239	10.52473	10.41042
99	9.66239	10.52473	10.41042
100	9.66239	10.52473	10.41042

The sample $Y = 50$ and generation is set at 100, crossover, mutation and recombination are parametrized at 0.2, 0.1, 0.5 respectively. Owing to the dynamic changes, there is a continuous parameter change over and above the different changes created by different functions discussed in this paper. The predictor is then subjected to these variegated conditions to determine from its vantage point the performance in dynamic environments. The error magnitude is weighed on absolute terms, where by the gap between the approximation and exactness is gradually reduced.

PMA provides better results in terms of error reduction. ε is reduced to the lowest level. Also, SA is subjected to the same functions in the experiment, to observe comparisons with other methods although it represents the local search paradigm. The functions include mutation, crossover, recombination and genetic that define candidate features and evolve them to better solutions. The lowest error obtained is in PMA as seen in Tables 1, 2 and 3. This can be attributed to nearest neighbor aspect which help to prevent errors in optimization.

7 Conclusion and Future Work

The cyclic change behaviors are important pointers for prediction. The presence of repeated patterns facilitate the process of prediction to significantly increase accuracy. This work proposed a PMA that estimates parameters in a dynamic environment to obtain good solutions. The parametrized changes as a result of the landscape changes are inherently nonlinear. The algorithm is applied to different nonlinear problem data sets and compared with genetic algorithm and simulated annealing. The proposed method produces superior results and generates the lowest error. As part of future work, multiple populations will be considered and further memory based approaches will be explored. More intelligent pattern recognition techniques for environment changes will also be incorporated in the PMA.

References

1. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Caltech Concurrent Computation Program, pp. 1–67 (1989)
2. Cassar, I.R., Titus, N.D.: An improved genetic algorithm for designing optimal temporal patterns of neural stimulation. *J. Neural Eng.* **14**, 1–15 (2017)
3. Forbes, N.: *Imitation of Life: How Biology is Inspiring Computing*, 1st edn. MIT Press, Cambridge (2004)
4. Dawkins, R.: Universal Darwinism. In: Bendall, D.S. (ed.) *Evolution from Molecules to Man*, pp. 2–16. Cambridge University Press, Cambridge (1983)
5. Dawkins, R.: *Memes: The New Replicators*, 2nd edn, pp. 188–300. Oxford University Press, Oxford (1989)
6. Merz, P., Freisleben, B.: Memetic algorithms for the traveling salesman problem. *J. Complex Syst.* **13**, 297–345 (1997)
7. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop schedulings. *IEEE Trans. Evol. Comput. Jpn.* **7**, 204–223 (2003)
8. Tang, J., Lim, M.H., Ong, Y.S.: Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems. *Soft Comput.* **11**, 873–888 (2007)
9. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. Evol. Comput.* **2**, 204–223 (2003)
10. Alkan, A., Ozcan, E.: Memetic algorithms for timetabling. In: *IEEE Proceedings of the 2003 Congress on Evolutionary Computation*, vol. 2, pp. 1796–1802 (2003)
11. Burke, E.K., Newall, J.P.: A multi-stage evolutionary algorithm for the timetable problem. *IEEE Trans. Evol. Comput.* **3**, 1085–1092 (1999)
12. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Keijzer, M. (ed.) *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 1201–1208. ACM (2006)

13. AbdAllah, A.M.F.M., Essam, D.L., Sarker, R.A.: Solving dynamic optimisation problem with variable dimensions. In: Dick, G., et al. (eds.) SEAL 2014. LNCS, vol. 8886, pp. 1–12. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13563-2_1
14. Zhou, Z., Ong, Y.S., Lim, M.H.: Memetic algorithm using multi-surrogates for computationally expensive optimization problems. *Soft Comput.* **11**, 957–971 (2007)
15. Mavrovouniotis, M., Yang, S.: A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Comput.* **15**, 1405–1425 (2011)
16. William, H., Krasnogor, N., Smith, J.E.: *Recent Advances in Memetic Algorithms*. Springer, Heidelberg (2005). <https://doi.org/10.1007/3-540-32363-5>
17. Weiss, G.: Timeweaver: a genetic algorithm for identifying predictive patterns in sequences of events. In: Spector, L. (ed.) *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 718–725 (1999)
18. Fang, K.T., Zhang, J.T.: A new algorithm for calculation of estimates of parameters of nonlinear regression modelings. In: *Proceedings of International Conference on Optimization Techniques and Applications*, pp. 1–8 (1995)
19. Rawlings, J.O., Pantula, S.G., Dickey, D.A.: *Applied Regression Analysis: A Research Tool*. Springer, New York (1998). <https://doi.org/10.1007/b98890>
20. Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E.: Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) *EMO 2007*. LNCS, vol. 4403, pp. 832–846. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70928-2_62
21. Hemert, J.V., Hoyweghen, C.V., Lukshandl, E., Verbeeck, K.: A futurist approach to dynamic environments. In: Branke, J., Back, T. (eds.) *Proceedings for Evolutionary Algorithms for Dynamic Optimization Problems at the Genetic and Evolutionary Computation Conference*, pp. 1–10 (2001)
22. Branke, J.: *Evolutionary Optimization in Dynamic Environments*, vol. 3. Springer, New York (2002). <https://doi.org/10.1007/978-1-4615-0911-0>
23. Weicker, K.: *Evolutionary algorithms and dynamic optimization problems*. Ph.D. thesis. University of Stuttgart, Germany (2003)
24. Stroud, P.D.: Kalman-extended genetic algorithm for search in nonstationary environments with noisy fitness evaluations. *IEEE Trans. Evol. Comput.* **5**, 66–77 (2001)
25. Karaman, A., Uyar, Ş., Eryiğit, G.: The memory indexing evolutionary algorithm for dynamic environments. In: Rothlauf, F., et al. (eds.) *EvoWorkshops 2005*. LNCS, vol. 3449, pp. 563–573. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-32003-6_59
26. Bosman, P.A.N., La Poutre, H.: Computationally intelligent online dynamic vehicle routing by explicit load prediction in an evolutionary algorithm. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *PPSN 2006*. LNCS, vol. 4193, pp. 312–321. Springer, Heidelberg (2006). https://doi.org/10.1007/11844297_32
27. Simoes, A., Costa, E.: Evaluating predictor’s accuracy in evolutionary algorithms for dynamic environments. In: *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pp. 883–890 (2009)
28. DeJong, K.: Evolutionary computation: a unified approach. *J. Evol. Comput.* **164–270** (2006)
29. Kapanoglu, M., Koca, I.O., Erdogmus, S.: Genetic algorithms in parameter estimation for nonlinear regression models: an experimental approach. *J. Stat. Comput. Simul.* **77**, 851–867 (2007)

30. Vollinger, U., Lehmann, E., Rainer, S.: Using memetic algorithms for the solution of technical problems. *Int. Sch. Sci. Res. Innov.* **3** (2009)
31. Bu, Z., Zheng, B.: Perspectives in dynamic optimization evolutionary algorithm. In: Cai, Z., Hu, C., Kang, Z., Liu, Y. (eds.) *ISICA 2010*. LNCS, vol. 6382, pp. 338–348. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16493-4_35
32. Wang, H., Wang, D., Yang, S.: A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Comput.* **13**, 763–780 (2008)
33. KJason, B.: *Clever Algorithms: Nature-Inspired Programming Recipes*. Lulu Enterprises, Morrisville (2011)
34. Hart, W.E., Krasnogor, N., Smith, J.E.: Memetic evolutionary algorithms. In: Hart, W.E., Smith, J.E., Krasnogor, N. (eds.) *Recent Advances in Memetic Algorithms. Studies in Fuzziness and Soft Computing*, vol. 166, pp. 3–27. Springer, Heidelberg (2005). https://doi.org/10.1007/3-540-32363-5_1
35. Gen, M., Cheng, R.: *Genetic Algorithms and Engineering Optimization*. Wiley, Hoboken (2000)
36. Eiben, A., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. *IEEE Comput. Soc.* 124–141 (1999)
37. Gerdes, I., Klawonn, F., Kruse, R.: *Evolutionre Algorithmen*. Vieweg Verlag, Wiesbaden (2004)
38. Schneburg, E., Heinzmann, F., Feddersen, S.: *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley, Boston (1994)
39. Weicker, K.: *Evolution are Algorithmen*, 1st edn. Teubner Verlag, Wiesbaden (2002)
40. Seber, G.A.F., Wild, C.J.: *Nonlinear Regression*. Wiley, Hoboken (2003)
41. Hongfeng, W., Dingwei, W., Shengxiang, Y.: A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Comput.* **13**, 763–780 (2009)
42. Weicker, K.: *Evolutionary algorithms and dynamic optimization problems*. Thesis. University of Stuttgart, Germany (2003)
43. Simoes, A., Costa, E.: Improving memory’s usage in evolutionary algorithms for changing environments. In: *IEEE Congress of Evolutionary Computation*, pp. 276–283 (2007)
44. Smyth, G.K.: Nonlinear regression. *Encycl. Environ.* **3**, 1405–1411 (2002)
45. Fang, K.T., Zhang, J.T.: A new algorithm for calculation of estimates of parameters of nonlinear regression modellings. *Acta Math. Appl. Sin.* **16**, 366–377 (1993)
46. National Institute of Standards and Technology: *Nonlinear Regression*. StRD (2018). Accessed http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml