View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by IntechOpen

We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the
TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

8

Searching for the best Points of interpolation using swarm intelligence techniques

Djerou L.[1], Khelil N.[1], Zerarka A.[1] and Batouche M.[2]
[1]*Labo de Mathématiques Appliquées, Université Med Khider*
[2]*Computer Science Department, CCIS-King Saud University*
[1]*Algeria,* [2]*Saudi Arabia*

1. Introduction

If the values of a function, ***f(x)***, are known for a finite set of x values in a given interval, then a polynomial which takes on the same values at these x values offers a particularly simple analytic approximation to ***f(x)*** throughout the interval. This approximating technique is called polynomial interpolation. Its effectiveness depends on the smoothness of the function being sampled (if the function is unknown, some hypothetical smoothness must be chosen), on the number and choice of points at which the function is sampled.
In practice interpolating polynomials with degrees greater than about 10 are rarely used. One of the major problems with polynomials of high degree is that they tend to oscillate wildly. This is clear if they have many roots in the interpolation interval. For example, a degree 10 polynomial with 10 real roots must cross the x-axis 10 times. Thus, it would not be suitable for interpolating a monotone decreasing or increasing function on such an interval.
In this chapter we explore the advantage of using the Particle Swarm Optimization (PSO) interpolation nodes. Our goal is to show that the PSO nodes can approximate functions with much less error than Chebyshev nodes.
This chapter is organized as follows. In Section 2, we shall present the interpolation polynomial in the Lagrange form. Section 3 examines the Runge's phenomenon; which illustrates the error that can occur when constructing a polynomial interpolant of high degree. Section 4 gives an overview of modern heuristic optimization techniques, including fundamentals of computational intelligence for PSO. We calculate in Subsection 4.2 the best interpolating points generated by PSO algorithm. We make in section 5, a comparison of interpolation methods. The comments and conclusion are made in Section 6.

2. Introduction to the Lagrange interpolation

If $x_0, x_1, \ldots x_n$ are distinct real numbers, then for arbitrary values $y_0, y_1, \ldots y_n$, there is a unique polynomial p_n of degree at most n such that $p_n(x_i) = y_i$ $(0 \leq i \leq n)$ (David Kincaid & Ward Cheney, 2002).
The Lagrange form looks as follows:

$$p_n\left(x\right) = y_0 l_0\left(x\right) + \ldots + y_n l_n\left(x\right) = \sum_{i=0}^{n} y_i l_i\left(x\right) \tag{1}$$

Such that coordinal functions can be expressed in the following

$$l_i\left(x\right) = \prod_{j=0, j\neq i}^{n} \frac{\left(x - x_j\right)}{\left(x_i - x_j\right)} \tag{2}$$

(David Kincaid & Ward Cheney, 2002), (Roland E. et al 1994).
are coordinal polynomials that satisfy

$$l_i(x_j) = \begin{cases} 1, & i=j \\ 0, & i\neq j \end{cases} \tag{3}$$

The Lagrange form gives an error term of the form

$$E_n\left(x\right) = f(x) - p_n\left(x\right) = \frac{f^{\left(n+1\right)}\left(c\right)}{\left(n+1\right)!}\phi_n\left(x\right) \tag{4}$$

Where

$$\phi_n\left(x\right) = \prod_{i=0}^{n}\left(x - x_i\right) \tag{5}$$

If we examine the error formula for polynomial interpolation over an interval [a, b] we see that as we change the interpolation points, we change also the locations c where the derivative is evaluated; thus that part in the error also changes, and that change is a "black hole" to us: we never know what the correct value of c is, but only that c is somewhere in the interval [a, b]. Since we wish to use the interpolating polynomial to approximate the Equation (4) cannot be used, of course, to calculate the exact value of the error f – P_n, since c, as a function of x is, in general, not known. (An exception occurs when the (n + 1)st derivative off is constant). And so we are likely to reduce the error by selecting interpolation points $x_0, x_1, \ldots, x_n$ so as to minimize the maximum value of product $\phi_n\left(x\right)$

The most natural idea is to choose them regularly distribute in [a, b].

3. Introduction to the Runge phenomenon and to Chebyshev approximations

3.1 Runge phenomenon

If x_k are chosen to be the points $x_k = a + k\dfrac{b-a}{n} \quad \left(k = 0, \ldots, n\right)$ (means that are equally spaced at a distance $2n + 1$ apart), then the interpolating polynomial $p_n(x)$ need not to converge uniformly on [a, b] as n $\longrightarrow \infty$ for the function f(x).

This phenomenon is known as the Runge phenomenon (RP) and it can be illustrated with the Runge's "bell" function on the interval [-5, 5] (Fig.1).

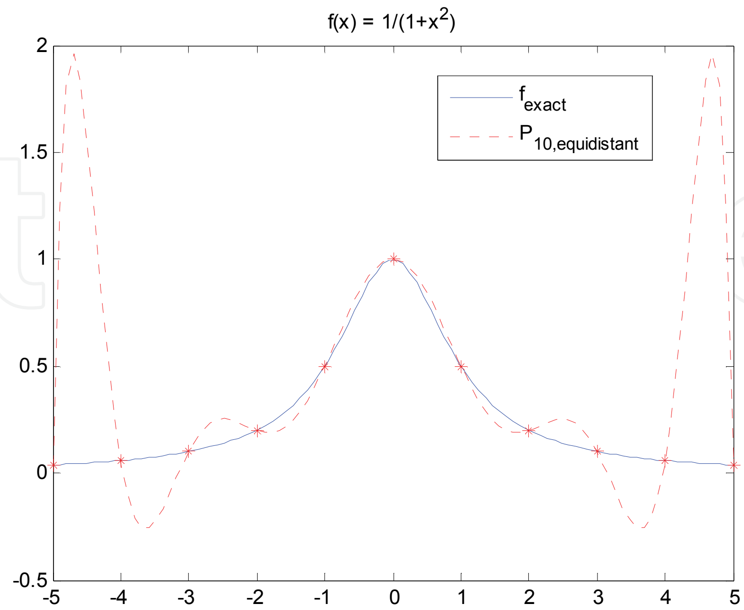


Figure 1. solid blue line: present Runge’s “bell” function. dots red line: present the polynomial approximation based on equally 11 spaced nodes

3.2 Chebyshev Nodes

The standard remedy against the RP is Chebyshev -type clustering of nodes towards the end of the interval (Fig.3).

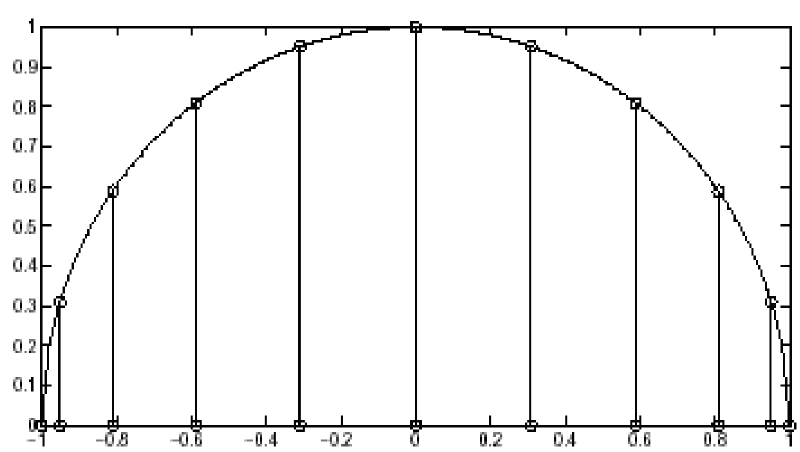


Figure 2. Chebyshev Point Distribution.

To do this, conceptually, we would like to take many points near the endpoints of the interval and few near the middle. The point distribution that minimizes the maximum value of product $\phi_n\left(x\right)$ is called the Chebyshev distribution, as shown in (Fig. 2). In the Chebyshev distribution, we proceed as follows:

1. Draw the semicircle on [a, b].

2. To sample $n + 1$ points, place n equidistant partitions on the arc.
3. Project each partition onto the x-axis: for j =0, 1… n

$$x_j = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(j\frac{\pi}{n}\right) \quad \text{for } j=0,1\ldots n \tag{6}$$

The nodes x_i that will be used in our approximation are:

Chebyshev nodes
-5.0000
-4.2900
-4.0251
-2.6500
-1.4000
0.0000
1.4000
2.6500
4.0451
4.2900
5.0000

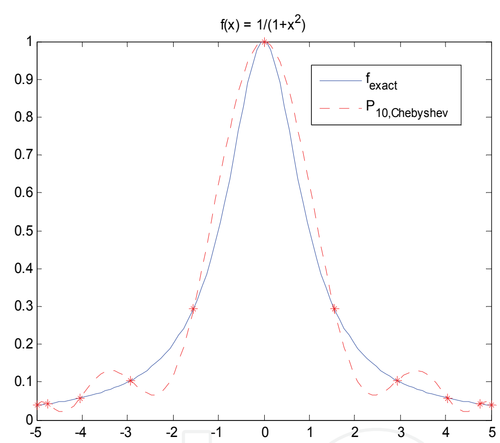


Figure 3. solid blue line: present Runge's "bell" function. dots red line: present the polynomial approximation based on 11 Chebyshev nodes

In this study, we have made some numerical computations using the particle swarm optimization to investigate the best interpolating points and we are showing that PSO nodes provide smaller approximation error than Chebyshev nodes.

4. Particle swarm optimization

4.1 Overview and strategy of particle swarm optimization

Recently, a new stochastic algorithm has appeared, namely 'particle swarm optimization' PSO. The term 'particle' means any natural agent that describes the `swarm' behavior. The PSO model is a particle simulation concept, and was first proposed by Eberhart and Kennedy (Eberhart, R.C. et al. 1995). Based upon a mathematical description of the social

behavior of swarms, it has been shown that this algorithm can be efficiently generated to find good solutions to a certain number of complicated situations such as, for instance, the static optimization problems, the topological optimization and others (Parsopoulos, K.E. et al., 2001a); (Parsopoulos, K.E.et al. 2001b); (Fourie, P.C. et al., 2000); (Fourie, P.C. et al., 2001). Since then, several variants of the PSO have been developed (Eberhart,R.C. et al 1996); (Kennedy, J. et al., 1998); (Kennedy, J. et al., 2001); (Shi, Y.H. et al. 2001); (Shi, Y. et al. 1998a.); (Shi, Y.H. et al., 1998b); (Clerc, M. 1999). It has been shown that the question of convergence of the PSO algorithm is implicitly guaranteed if the parameters are adequately selected (Eberhart, R.C. et al.1998); (Cristian, T.I. 2003). Several kinds of problems solving start with computer simulations in order to find and analyze the solutions which do not exist analytically or specifically have been proven to be theoretically intractable.
The particle swarm treatment supposes a population of individuals designed as real valued vectors - particles, and some iterative sequences of their domain of adaptation must be established. It is assumed that these individuals have a social behavior, which implies that the ability of social conditions, for instance, the interaction with the neighborhood, is an important process in successfully finding good solutions to a given problem.
The strategy of the PSO algorithm is summarized as follows: We assume that each agent (particle) i can be represented in a N dimension space by its current position $x_i = \left(x_{i1}, x_{i2}, \ldots, x_{iN}\right)$ and its corresponding velocity. Also a memory of its personal (previous) best position is represented by, $p = \left(p_{i1}, p_{i2}, \ldots, p_{iN}\right)$ called (pbest), the subscript i range from 1 to s, where s indicates the size of the swarm. Commonly, each particle localizes its best value so far (pbest) and its position and consequently identifies its best value in the group (swarm), called also (sbest) among the set of values (pbest).
The velocity and position are updated as

$$v_{ij}^{k+1} = w_j v_{ij}^k + c_1 r_1^k \left[(pbest)_{ij}^k - x_{ij}^k\right] + c_2 r_2^k \left[(sbest)_{ij}^k - x_{ij}^k\right] \tag{7}$$

$$x_{ij}^{k+1} = v_{ij}^{k+1} + x_{ij}^k \tag{8}$$

where are the position and the velocity vector of particle i respectively at iteration k + 1, c_1 et c_2 are acceleration coefficients for each term exclusively situated in the range of 2--4, w_{ij} is the inertia weight with its value that ranges from 0.9 to 1.2, whereas r_1, r_2 are uniform random numbers between zero and one. For more details, the double subscript in the relations (7) and (8) means that the first subscript is for the particle i and the second one is for the dimension j. The role of a suitable choice of the inertia weight w_{ij} is important in the success of the PSO. In the general case, it can be initially set equal to its maximum value, and progressively we decrease it if the better solution is not reached. Too often, in the relation (7), w_{ij} is replaced by w_{ij} / σ, where σ denotes the constriction factor that controls the velocity of the particles. This algorithm is successively accomplished with the following steps (Zerarka, A. et al., 2006):

1. Set the values of the dimension space N and the size s of the swarm (s can be taken randomly).
2. Initialize the iteration number k (in the general case is set equal to zero).
3. Evaluate for each agent, the velocity vector using its memory and equation (7), where pbest and sbest can be modified.
4. Each agent must be updated by applying its velocity vector and its previous position using equation [8].
5. Repeat the above step (3, 4 and 5) until a convergence criterion is reached.

The practical part of using PSO procedure will be examined in the following section, where we'll interpolate Runge's "bell", with two manners; using Chebyshev interpolation approach and PSO approach, all while doing a comparison.

4.2 PSO distribution

So the problem is the choice of the points of interpolation so that quantity $\phi_n\left(x\right)$ deviates from zero on [a, b] the least possible.

Particle Swarm Optimization was used to find the global minimum of the maximum value of product $\phi_n\left(x\right)$, where very x is represented as a particle in the swarm.

The PSO parameter values that were used are given in Table 1.

Parameter	Setting
Population size	20
Number of iterations	500
C1 and C2	0.5
Inertial Weight	1.2 to 0.4
Desired Accuracy	10-5

Table 1. Particle Swarm Parameter Setting used in the present study

The best interpolating points x generated by PSO algorithm for polynomial of degree 5 and 10 respectively for example are:

Chebyshev	Points generated with PSO
-5.0000	-5.0000
-3.9355	-4.0451
-2.9041	-1.5451
0.9000	1.5451
3.9355	4.0451
5.0000	5.0000

Table 2 Polynomial of degree 5

Chebyshev	Points generated with PSO
-5.0000	-5.0000
-4.2900	-4.7553
-4.0251	-4.0451
-2.6500	-2.9389
-1.4000	-1.5451
0.0000	-0.0000
1.4000	1.5451
2.6500	2.9389
4.0451	4.0451
4.2900	4.7553
5.0000	5.0000

Table 3. Polynomial of degree 10

5. Comparison of interpolation methods

How big an effect can the selection of points have? Fig. 4 and Fig. 5 shows Runge's "bell" function interpolated over [-5, 5] using equidistant points, points selected from the Chebyshev distribution, and a new method called PSO. The polynomial interpolation using Chebyshev points does a much better job than the interpolation using equidistant points, but neither does as well as the PSO method.

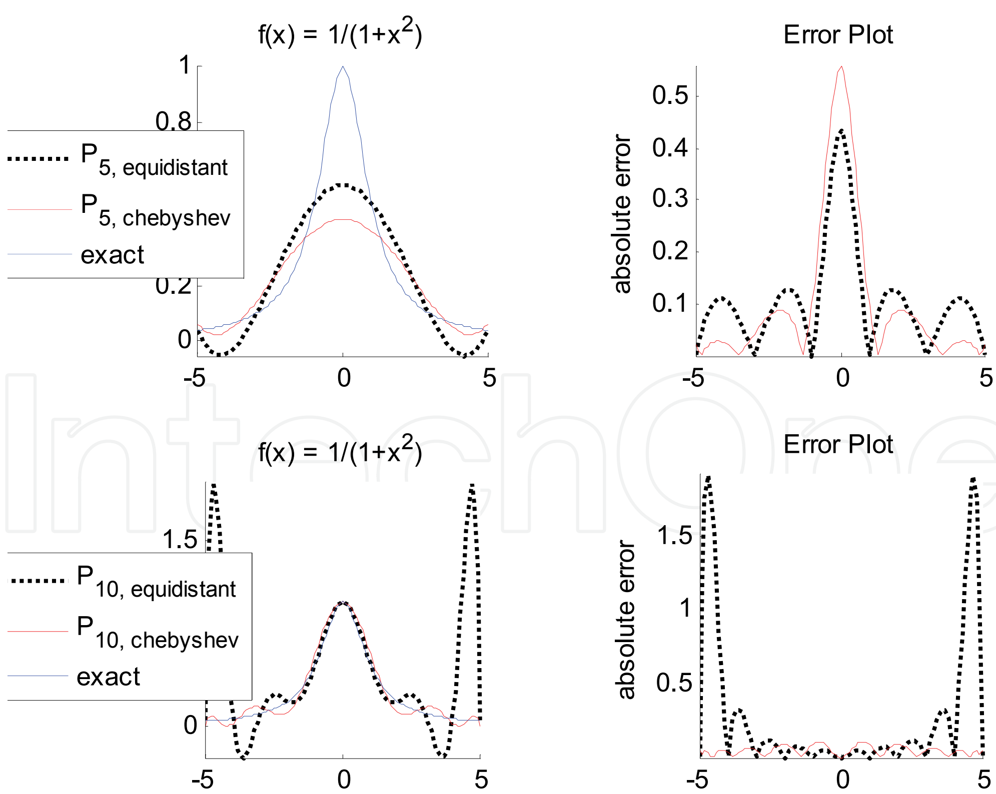


Figure 4. Comparison of interpolation polynomials for equidistant and Chebyshev sample points

Comparing Fig. 4, we see that the maximum deviation of the Chebyshev polynomial from the true function is considerably less than that of Lagrange polynomial with equidistant nodes. It can also be seen that increasing the number of the Chebyshev nodes—or, equivalently, increasing the degree of Chebyshev polynomial—makes a substantial contribution towards reducing the approximation error.

Comparing Fig. 5, we see that the maximum deviation of the PSO polynomial from the true function is considerably less than that of Chebyshev polynomial nodes. It can also be seen that increasing the number of the PSO nodes—or, equivalently, increasing the degree of PSO polynomial—makes a substantial contribution towards reducing the approximation error.

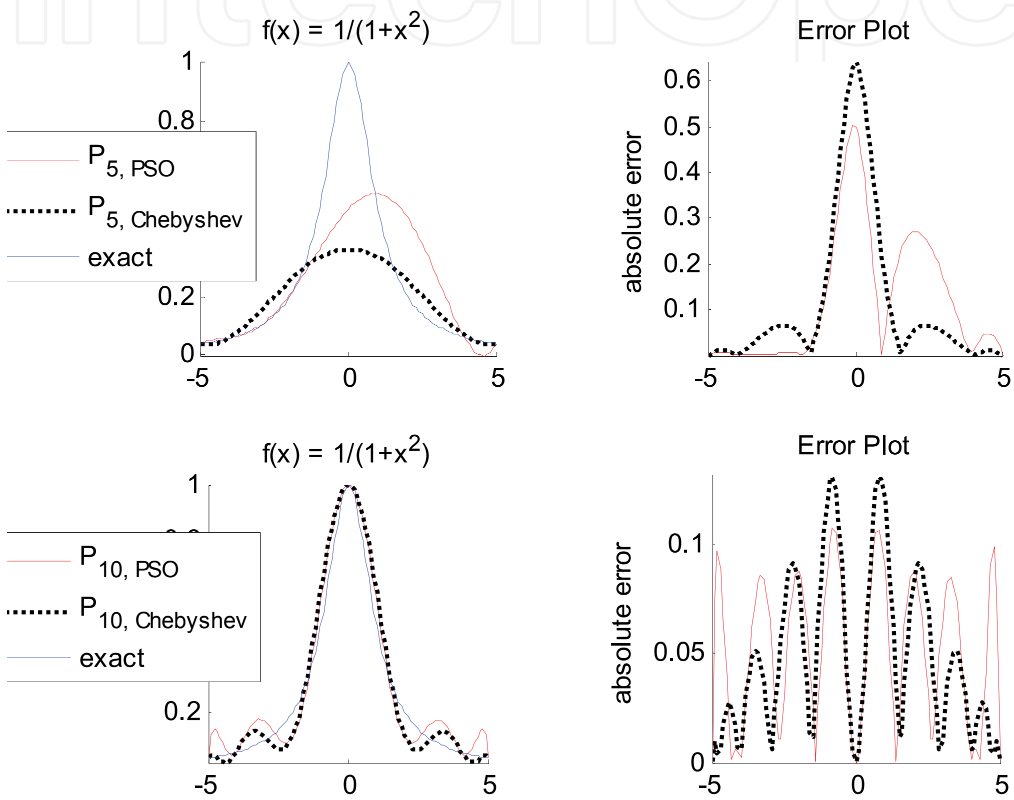


Figure 5. Comparison of interpolation polynomials for PSO and Chebyshev sample points

In this study we take as measure of the error of approximation the greatest vertical distance between the graph of the function and that of the interpolating polynomial over the entire interval under consideration (Fig. 4 and Fig. 5).

The calculation of error gives

Degree	Error points equidistant	Error points Chebychev	Error points PSO
5	0.4327	0.6386	0.5025
10	1.9156	0.1320	0.1076
15	2.0990	0.0993	0.0704
20	58.5855	0.0177	0.0131

Table 2. The error

6. Conclusion

The particle swarm optimization is used to investigate the best interpolating points. Some good results are obtained by using the specific PSO approach. It is now known that the PSO scheme is powerful, and easier to apply specially for this type of problems. Also, the PSO method can be used directly and in a straightforward manner. The performance of the scheme shows that the method is reliable and effective.

7. References

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, *Proceedings of the 1999 IEEE Congress on Evolutionary Computation, pp.1951 – 1957*, Washington DC.

Cristian, T.I. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection, *Information Processing Letters,* Vol. 85, No. 6, pp.317--325.

David Kincaid and Ward Cheney, (2002). *Numerical Analysis: Mathematics of Scientific Computing*. Brooks/Cole.

Eberhart, R.C. and Kennedy, J. (1995). A new optimizer using particles swarm theory', *Sixth International Symposium on Micro Machine and Human Science,* pp.39--43, *Nagoya*, Japan.

Eberhart, R.C. and Shi, Y. (1998). Parameter selection in particle swarm optimization, in Porto, V.W.,

Eberhart, R.C. et al (1996). Computational Intelligence PC Tools, *Academic Press Professional*, Boston.

Fourie, P.C. and Groenwold, A.A. (2000). Particle swarms in size and shape optimization', *Proceedings of the International Workshop on Multi-disciplinary Design Optimization*, August 7--10, pp.97—106, Pretoria, South Africa.

Fourie, P.C. and Groenwold, A.A. (2001). Particle swarms in topology optimization', *Extended Abstracts of the Fourth World Congress of Structural and Multidisciplinary Optimization*, June 4--8, pp.52, 53, Dalian, China.

Hammer, R. Et al (1995). Numerical Toolbox for Verified Computing I, *Springer Verlag*, Berlin.

Kennedy, J. 1998. The behavior of particles, *Evol. Progr.*, Vol. VII, pp.581-587.

Kennedy J. and Eberhart, R.C, (1995). Particle swarm optimization, *Proc. IEEE Int. Conf. Neural Networks, Piscataway*, NJ, pp.1942--1948, USA.

Kennedy, J. and Eberhart, R.C. (2001). Swarm Intelligence, *Morgan Kaufmann Publishers*, San Francisco.

Kennedy, J. and Spears, W.M. (1998). Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator, *Proceedings of the (1998) IEEE International Conference on Evolutionary Computation*, Anchorage, May 4--9, Alaska.

Kulisch, U. and Miranker, W.L. (1983). A New Approach to Scientific Computation, Academic Press, New York.

L. N. Trefethen. Spectral Methods in Matlab. *SIAM*, (2000). 9, Philadelphia

L. Djerou, M. Batouche, N. Khelil and A.Zerarka, *(2007)*. Towards the best points of interpolation using Particles swarm optimization approach*, in proceedings of IEEE Congress of Evolutionary Computing, CEC 2007, pp. 3211-3214, Singapore.*

Maron, M. and Lopez, R. (1991). Numerical Analysis, *Wadsworth Publishing Company*, Belmont, California.

Parsopoulos, K.E. and Vrahatis, M.N. (2001). `Modification of the particle swarm optimizer for locating all the global minima', in Kurkova, V. et al. (Eds.): *Artificial Neural Networks and Genetic Algorithms, Springer*, pp.324--327, New York.

Parsopoulos, K.E. et al, (2001a). Objective function stretching to alleviate convergence to local minima, *Nonlinear Analysis TMA*, Vol. 47, pp.3419--3424.

Parsopoulos, K.E.et al (2001b). Stretching technique for obtaining global minimizers through particle swarm optimization, *Proceedings of the PSO Workshop*, pp.22--29, Indianapolis, USA.

Roland E. Larson, Robert P. Hostetler, Bruch H. Edwards and David E. Heyd, (1994)., *Calculus with Analytic Geometry*. D. C. Heath and Company.

Saravanan, N., Waagen, D. and Eiben, A.E. (Eds.): *Lecture Notes in Computer Science-Evolutionary Programming VII, Springer*, Vol. 1447, pp.591--600.

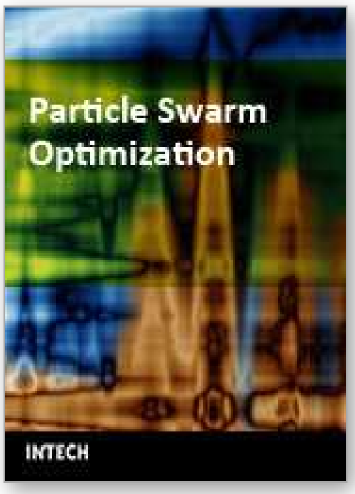
Shi, Y. and Eberhart, R.C. (1998a). A modified particle swarm optimizer, *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, May 4—9, Anchorage, Alaska.

Shi, Y.H. and Eberhart, R.C. (1998b). Parameter selection in particle swarm optimization, Evolutionary Programming VII, *Lecture Notes in Computer Science*, pp.591--600.

Shi, Y.H. and Eberhart, R.C. (2001). Fuzzy adaptive particle swarm optimization', *IEEE Int. Conf. on Evolutionary Computation*, pp.101--106.

Zerarka, A., Khelil, N. (2006). A generalized integral quadratic method: improvement of the solution for one dimensional Volterra integral equation using particle swarm optimization, *Int. J. Simulation and Process Modeling*, Vol. 2, Nos. 1/2, pp.152-163.

IntechOpen



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0

Hard cover, 476 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling.PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Djerou L., Khelil N., Zerarka A. and Batouche M. (2009). Searching for the Best Points of Interpolation Using Swarm Intelligence Techniques, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from: http://www.intechopen.com/books/particle_swarm_optimization/searching_for_the_best_points_of_interpolation_using_swarm_intelligence_techniques

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.