A Modified Particle Swarm Optimizer Using an Adaptive Dynamic Weight Scheme

Shu-Kai S. Fan and Ju-Ming Chang

Department of Industrial Engineering and Management
Yuan Ze University, Taoyuan County, Taiwan 320, Republic of China
simonfan@saturn.yzu.edu.tw

Abstract. Particle swarm optimization (PSO) is a stochastic, population-based optimization technique that is inspired by the emigrant behavior of a flock of birds searching for food. In this paper, a nonlinear function of decreasing inertia weight that adapts to current performance of PSO search is presented. Meanwhile, a dynamic mechanism to adjust decrease rates is also suggested. Through the experimental study, the new PSO algorithm with adaptive dynamic weight scheme is compared to the exiting models in terms of various benchmark functions. The computational experience shows some great promise.

Keywords: particle swarm optimization (PSO), dynamic inertia weight, population-based optimization technique.

1 Introduction

Many population-based search algorithms take inspiration from some natural system involving groups of interacting individuals, such as genetic algorithm (GA) [1], ant colony optimization (ACO) [2] and particle swarm optimization (PSO) [3]. Metaheuristics other than population-based approaches include simulated annealing (SA), tabu search (TS), variable neighborhood search (VNS), among others. The population-based evolutionary computation technique is operated through cooperation and competition among the potential solutions. One of prominent merits owned by the population-based techniques is the ability to anchor an approximate optimum solution quite effectively when applied to complicated optimization problems. PSO was first introduced by Eberhart and Kennedy [3] in 1995. It is noted for dealing with optimization in continuous, multimodal search spaces. Although PSO can produce remarkable performance for many practical problems of continuous optimization, the primitive PSO still suffers the consequences of slow convergence or convergence failure if the problem size is getting large [4]. Researchers from different research fields have been devoted to improving the performance of the original PSO. To date, the most well-known modification is the one incorporating an extra parameter called *inertia weight* into the original PSO algorithm. The objective is to balance the global and local exploration capability [5]. The inertia weight is deliberately designed to adjust the current velocity that resembles a local search behavior. This can be visualized by tracking the search locus of the best particle *en route*. By including the inertia weight together with sharing information among swarm, the particles tend to

V.G. Duffy (Ed.): Digital Human Modeling, HCII 2007, LNCS 4561, pp. 56–65, 2007.

expand their search ability as much as they can; that is, they become more capable of exploring the new areas (namely, global search ability). Combining both the local and global search would definitely benefit solving highly nonlinear, multimodal optimization problems.

The inertia weight ω is employed as to control the impact of the previous history of velocities on the current velocity, therefore influencing the trade-off between global and local exploration of particle search. A large value of inertia weight facilitates global exploration, while a small inertia weight tends to facilitate local exploration that helps to fine-tune the current neighborhood search. A suitable selection of inertia weight can provide an appropriate balance between global and local exploration abilities, so less computation effort is anticipated for convergence. Shi and Eberhart [6] claimed that a reasonable choice of ω should decrease gradually while the swarm search progresses. As a general remark, they opined that a better performance would be expected if the inertia weight is chosen time-varying, linearly decreasing with iteration, rather than just using a constant value. Their argument is supported by examining a single case study. It was inferred that PSO should start with a high inertia weight for coarse global exploration and the inertia weight should linearly decrease to facilitate finer local explorations in later iterations. This should help PSO to approach the optimum efficiently.

Based on the hypothesis that dynamic adjustment of inertia weight is necessary for PSO, some improvements have been reported in the literature. Most recently, Chatterjee and Siarry [7] proposed a PSO model with dynamic adaptation that concentrates on the adjustability of the decreasing rate for the inertia weight ω. In their new approach, a set of exponential-like curves are used to create a complete set of free parameters for any given problem, saving the data from a tedious, trial-and-error-based approach to determine them for each specific problem. As evidenced by the experimental results, their approach can successfully solve several benchmark functions and attain better convergence than existing inertia weight PSO algorithms. Nonetheless, an obvious shortcoming is that the maximum number of iterations to define the decreasing inertia weight needs to be specified in advance.

2 The Particle Swarm Optimization (PSO)

PSO is a swarm intelligence algorithm that emulates a flock of birds searching over the solution landscape by sampling points and the swarm converges on the most promising regions. A particle moves through the solution space along a trajectory defined by its velocity, the draw to return to a previous promising search area, and an attraction towards the best area discovered by its neighbors. The force behind the convergence of a swarm is of social pressure, applied through the interplay between the particles in flight.

2.1 The Static PSO

Eberhart and Kennedy [3] developed a ground-breaking algorithm through simulating social behavior, which is called particle swarm optimization (PSO). In a particle swarm, each individual is influenced by its closest neighbors and each particle is a possible solution to the problem being optimized. PSO formulae define each particle as a potential solution in N-dimensional space, with each indicated as X_{id}. The update of the ith particle in the swarm of size s is accomplished according to the equations (1) and (2), as expressed by:

$$V_{id} = V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{gd} - X_{id}) ; \tag{1}$$

$$X_{id} = X_{id} + V_{id}, \text{ for } d = 1,\ 2, \ldots,\ \text{N},\ i = 1,\ 2, \ldots,\ s . \tag{2}$$

Equation (1) calculates a new velocity for each particle based on its previous velocity V_{id}, and the previous best particle P_{id}, and the global best particle P_{gd}. Equation (2) updates each particle's position in the solution space. The two random numbers r_1 and r_2 are independently generated, and c_1 and c_2 are the learning factors determining the relative influence of the cognitive and social components, respectively. The value of each component in the velocity vector can be clamped to the range $[-v_{\max}, v_{\max}]$ so as to reduce the likelihood of particles leaving the search space. The value of $v_{\max}$ only limits the maximum distance that a particle can move in every iteration. Since the pioneering PSO work done by Kennedy and Eberhart [3], abundant research has been working on improving its performance in various ways, thereby publishing many interesting modified PSO versions. One of the most prominent PSO models [6] introduces a parameter called the inertia weight ω into the original PSO algorithm, and therefore the velocity update rule becomes:

$$V_{id} = \omega V_{id} + c_1 r_1 (P_{id} - X_{id}) + c_2 r_2 (P_{gd} - X_{id}) . \tag{3}$$

The parameter of inertia weight is used to balance the global and local search abilities. A large inertia weight facilitates global search; a small inertia weight facilitates local search.

2.2 The Proposed Dynamic Adaptive PSO Algorithm

In an empirical study on PSO [6], Shi and Eberhart claimed that a linearly decreasing inertia weight could improve local search towards the end of a run, rather than using a constant value throughout. They supported their advocate with a single case study. Furthermore, it was also asserted that the swarm may fail to converge to a good solution by restricting the global search ability too much. It implies that a large value of the inertia weight could make the particles own more global search ability to explore new search areas; on the other hand, a small inertia weight produces less variation in velocity to perform a better local search. In this light, a decreasing function for the dynamic inertia weight can be devised in the following

$$\omega = \left(\text{iter}_{\max} - \text{iter}_{\text{cur}}\right)\left(\frac{\omega_{\text{initial}} - \omega_{\text{final}}}{\text{iter}_{\max}}\right) + \omega_{\text{final}} , \tag{4}$$

where ω_{initial} and ω_{final} represent the initial and final inertia weights respectively at the start of a given run, $\text{iter}_{\max}$ the maximum number of iterations in a offered run, and $iter_{cur}$ the current iteration number at the present time step. More recently, a different version of dynamic inertia weight was proposed by Chatterjee and Siarry [7]. They presented a different nonlinear function that modulates the inertia weight with time for improving the performance of the PSO algorithm. The main modification is the determination of the inertia weight as a nonlinear function of the current iteration number at each time step. The dynamic function of ω was modified as follows:

$$\omega = \left(\frac{iter_{max} - iter_{cur}}{iter_{max}} \right)^n \left(\omega_{inital} - \omega_{final} \right) + \omega_{final} . \tag{5}$$

Through the study of the nonlinear modulation parameter n, they also derived a reasonable set of choice for the free parameters. In addition, better results have been obtained for several benchmark problems while the nonlinear modulation n=1.2 was opted. The modified PSO is initially with a high inertia weight to explore new search areas. Later, with gradually decreasing inertia weights following different paths of different values of n, the final inertia weight is reached at the maximum number of iterations. In this paper, a new version of dynamic, adaptive inertia weight is proposed, which will be detailed in the next section.

2.3 Adaptive PSO Model with Dynamic Inertia Weight

With the decreasing inertia weight model, once the swarm finds the local area where the optimal solution resides, the particles would be attracted toward the neighborhood. The trajectory of each particle is pinned to its best ever visited solution. As new, better solutions are discovered, each particle updates its current solution for seeking out even better solutions. As such, the search spread is getting tight step by step. Consider a situation that the final destination is only a "local" optimal solution in that area. When converged, the decreasing inertia weight must turn out to be an extremely small value and the particles hardly escape from the local optimal area. A reasonable

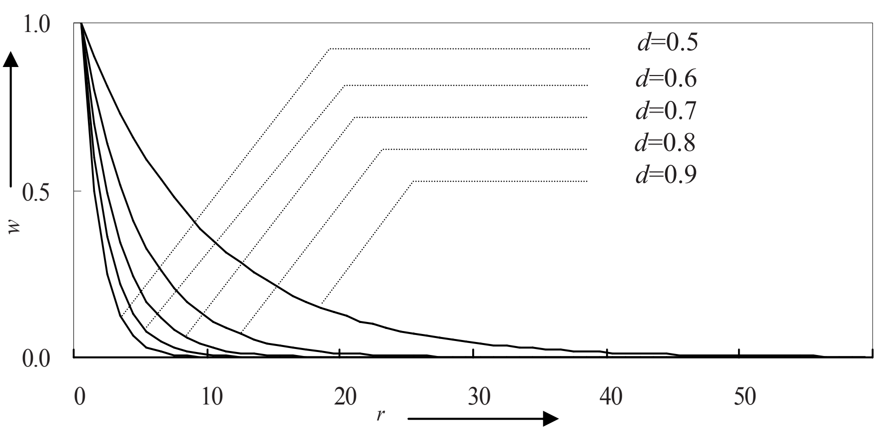

Fig. 1. Inertia weights with different values of the decrease rate d

strategy for improvement can be envisioned that if a particle finds a better solution then more energy (*i.e.*, weight) is given onto the current velocity to speed up exploitation, and *vice versa*. Under such circumstances, extended search and/or local refinement can be realized.

To achieve the foregoing concept, the present paper proposes a novel nonlinear function regulating inertia weight adaptation with a dynamic mechanism for enhancing the performance of PSO algorithms. The principal modification is the determination of the inertia weight through a nonlinear function at each time step. The nonlinear function is given by

$$\omega = (d)^r \, \omega_{initial} , \tag{6}$$

where d is the decrease rate ranging between 1.0 and 0.1, and r the dynamic adaptation parameter depending on the following rules for successive adjustment at each time step. For a minimization case, it follows

$$\text{if } f\left(P_{gd-new}\right) < f\left(P_{gd-old}\right) \text{ then } r \leftarrow r - 1 ; \tag{7}$$

$$\text{if } f\left(P_{gd-new}\right) \geq f\left(P_{gd-old}\right) \text{ then } r \leftarrow r + 1 , \tag{8}$$

where P_{gd-new} and P_{gd-old} denote the global best solutions at current and previous time steps, respectively. This mechanism wishes to make particles fly more quickly toward the potential optimal solution, and then through decreasing inertia weight to perform local refinement around the neighborhood of the optimal solution. Fig. 1 illustrates decreasing inertia weight plots with different decrease rates.

3 Behavior of Proposed Adaptive Dynamic PSO

To study the behavior of the proposed PSO model, the Sphere function was applied, which has a single minimum with adjustable problem size n, as defined by

$$\text{Sphere}_n(x) = \sum_{i=1}^{n} x_i^2 . \tag{9}$$

The Sphere function has the objective value 0 at its global minimum $x = (0, \ldots, 0)$. In this experiment of parameter analysis, the 10-dimensional Sphere

Table 1. Performance returned by the proposed PSO model for the Sphere function using various decrease rates (d)

Decrease rate (d)	Success rate (%)	Average function error	Iterations
0.5	100	9.9518e–7	519
0.6	100	9.8836e–7	362
0.7	100	8.6420e–7	281
0.8	100	7.4526e–7	197
0.9	100	7.1711e–7	137
0.95	100	9.3543e–7	153

model is used, the swarm size is $s = 20$, and the initial inertia weight is $\omega_{initial} = 1$. The search domain is restricted within $[-5.12,\ 5.12]$. The problem was optimized for 50 independent runs, and all particles in each dimension can have a maximum allowable velocity which is confined to the full range in a given dimension.

For evaluation purposes, it is necessary to give a formal definition of successful minimization, an instance to be considered as a success if the following inequality holds:

$$\left|\mathrm{FOBJ}_{\text{Proposed-PSO}} - \mathrm{FOBJ}_{\text{ANAL}}\right| < \varepsilon_{rel} \times \mathrm{FOBJ}_{\text{ANAL}} + \varepsilon_{abs}\,, \tag{10}$$

where the relative tolerance $\varepsilon_{rel} = 10^{-4}$ and the absolute tolerance $\varepsilon_{abs} = 10^{-6}$ are chosen. $\mathrm{FOBJ}_{\text{Proposed-PSO}}$ is referred to as the best function value achieved by the proposed algorithm, and $\mathrm{FOBJ}_{\text{ANAL}}$ is referred to as the exact analytical global objective. The optimization results are listed in Table 1, containing the average function error, the success rate over 50 runs, and the average iterations required. It can be observed from the table that the optimum decrease rate for the Sphere function is 0.9, requiring minimum iteration number 137 for convergence.

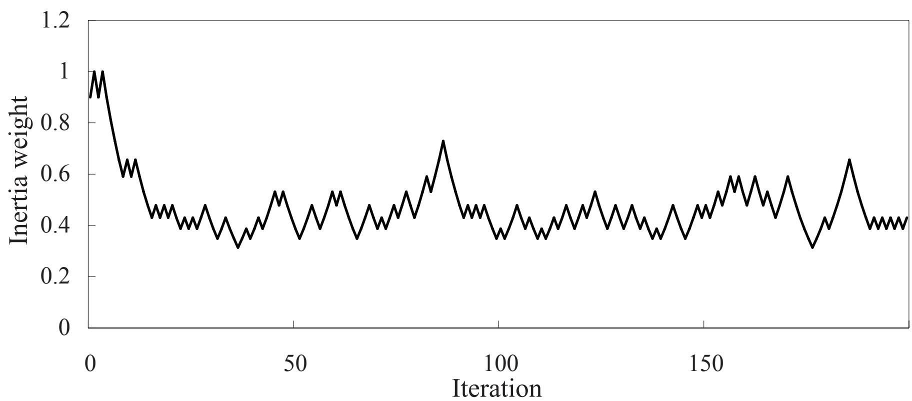

Fig. 2. Adaptive inertia weight profile using decrease rate d = 0.9 when solving the 10-D Sphere function

Fig. 2 pictorially shows the trend of adaptive inertia weights in a single optimization run of the 10-D Sphere function. Obviously, most of inertia weights fall from 0.4 to 0.7. This range of ω was also suggested in [8-9]. Hence, a choice of using d = 0.9 can be considered a rule of thumbs for general cases. In the figure, the spike indicates extended exploration conducted in the favorable search direction that has been found; a zigzagging that hovers around a specific inertia weight level indicates local refinement. The parameter setting recommended here will be used for subsequent experimental comparisons to be presented in section 4.

4 Performance of Proposed PSO Model

This section presents the performance of the proposed PSO model on 8 benchmark functions with 2 to 10 dimensions. They consist of 2-dimensional Rosenbrock, Quadric, Ackley, Rastrigin, Griewank, and Zakharov functions, and two higher dimensional problems, *i.e.*, 5- and 10-dimensional Zakharov functions. These benchmark functions have the global optimum objective of zero. Furthermore, the optimization results were also compared with the other two dynamic PSO models using different decreasing inertia weight functions. These two schemes are the nonlinear inertia weight PSO [7] and linear inertia weight PSO models [6]. The proposed PSO model uses the parameter setting: $\omega_{initial} = 1$ and $d = 0.9$. Note again that all experiments were run 50 times with 20 particles for 500 iterations. $v_{\max}$ is clamped to the full range in a given dimension for each benchmark function. The results reported are the averages calculated from all 50 runs. Table 2 shows the average objective function errors returned by using three dynamic inertia weight PSO versions. The proposed PSO model performed best (in solution precision) for all test problems except the Rastrigin function for which an equally good result was yielded by the other two schemes.

Table 2. Average objective function errors computed over 50 runs by three PSO models for 2 to 10 dimensional benchmark functions

Function	Decreasing Inertia Weight Method		
	Proposed Adaptive *w* PSO	Nonlinear *w* PSO [7]	Linear *w* PSO [6]
Rosenbrock	1.329985e−10	3.151920e−06	1.510120e−05
Quadric	7.420908e−101	2.534761e−98	2.756407e−91
Ackley	5.887218e−16	1.835229e−14	5.152140e−13
Rastrigin	0.000000e+00	0.000000e+00	0.000000e+00
Griewank	1.194600e−13	2.764728e−06	1.067861e−04
Zakhavior_2	4.413094e−114	3.030987e−88	1.588612e−80
Zakhavior_5	2.821841e−43	1.156531e−38	1.464352e−32
Zakhavior_{10}	1.520226e−12	1.347304e−07	5.346056e−04

To gain a better understanding of convergence property, Fig. 3 plots the logarithm of function error achieved by three PSO models versus iteration for the 10-D Zakhavior problem. The proposed PSO model exhibits an excellent, monotone convergence speed.

To further verify the proposed PSO model, 5 benchmark functions of higher dimension were tested in this experiment. The problem size of 30 was for every benchmark problems, and the performance was evaluated with 2000 iterations in each run. The optimization results were reported according to three different swarm sizes, 20, 40 and 60 particles. Table 3 presents the comparison results. Likewise, the proposed PSO model outperformed the other two models almost in every cases.

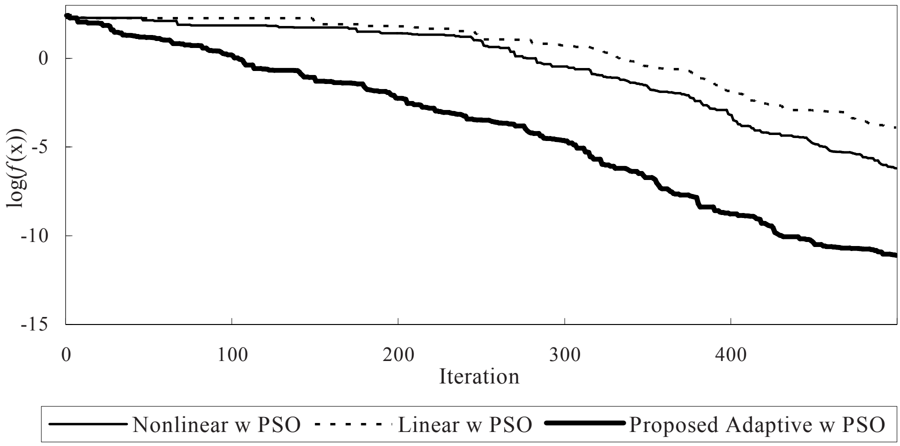

Fig. 3. Objective improvement of three PSO models for the 10-D Zakhavior function

Multimodal functions such as Ackley, Rastrigin and Griewank functions with many local minima positioned on a regular grid would pose a serious challenge for the algorithms. Yet, the proposed model still converged faster than the other two models, as can be seen from Fig. 4 for the Rastrigin function. The nonlinear and linear inertia weight PSO models appeared to get trapped in two different local minima.

Table 3. Average objective function errors returned by three PSO models for the 30-D problems using 3 different swarm sizes

Function	Pop.	Decreasing Inertia Weight Method		
		Proposed Adaptive *w* PSO	Nonlinear *w* PSO	Linear *w* PSO
Rosenbrock	20	8.386216e+01	1.361161e+02	1.538252e+02
	40	6.345287e+01	8.074967e+01	9.734088e+01
	60	3.785684e+01	6.664381e+01	8.009522e+01
Quadric	20	1.254913e+03	1.684360e+03	2.263458e+03
	40	1.297764e+02	6.945127e+02	8.299236e+02
	60	4.813780e+01	4.139014e+02	6.293952e+02
Ackley	20	2.450348e+00	2.775425e+00	3.825188e+00
	40	1.109738e+00	1.666886e+00	2.701517e+00
	60	9.706417e–01	1.103057e+00	1.275164e+00
Rastrigin	20	5.366067e+01	5.897047e+01	8.184283e+01
	40	3.676980e+01	3.634060e+01	7.364739e+01
	60	1.008213e+01	2.848330e+01	6.390452e+01
Griewank	20	8.849953e–02	1.344367e–01	9.558794e–01
	40	2.460268e–02	3.195703e–02	4.928395e–02
	60	9.857285e–03	2.956801e–02	9.358511e–02

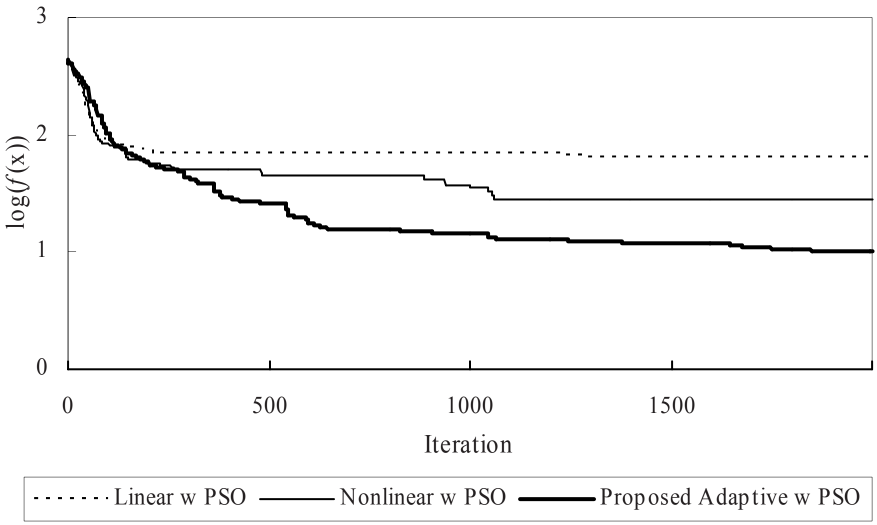

Fig. 4. Objective reduction versus iteration for the 30-D Rastrigin function

5 Conclusion

This paper presented a new adaptive scheme which employs a nonlinear function to provide dynamic inertia weights for PSO. This resulted in a significant improvement in performance, especially in terms of the solution quality and convergence speed. The success of the proposed model can be attributed mainly to an adequate selection of decrease rate. Meanwhile, a dynamic mechanism to adjust decrease rates is also suggested. A major difference from Chatterjee and Siarry's model is that the maximum number of iterations for designing the decreasing inertia weight need not be known beforehand. Through a series of experimental studies, the new PSO model was compared to two exiting models in terms of various benchmark functions. The computational experience shows some great promise.

References

1. Goldberg, D.E.: Genetic Algorithms in Search Optimization, and Machine Learning. Addison-Welsey, Reading MA (1989)
2. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of CAL'91—European Conference on Artificial Life, pp. 134–142. Elsevier Publishing, Paris, France (1998)
3. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth international Symposium on Micro Machine and Human Science. pp. 39–43, IEEE Service Center, Piscataway, NJ, Nagoya, Japan (1995)

4. Angeline, P.J.: Evolutionary optimization versus particle swarm optimization: philosophy and performance differences. In: Porto, V.W., Waagen, D. (eds.) Evolutionary Programming VII. LNCS, vol. 1447, pp. 601–610. Springer, Berlin (1998)
5. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 69–73. IEEE Press, Piscataway NJ (1998)
6. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: Evolutionary Programming VII. Proc. EP98, pp. 591–600. Springer, New York (1998)
7. Chatterjee, A., Siarry, P.: Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. Computers & Operations Research 33, 859–871 (2006)
8. Clerc, M.: Discrete Particle Swarm Optimization Illustrated by the Traveling Salesman Problem. Technical report (2000)
9. Parsopoulos, K.E., Vrahatis, M.N.: Initializing the particle swarm optimizer using the nonlinear simplex method. In: Grmela, A., Mastorakis, N.E. (eds.): Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, pp. 216–221, WSEAS Press (2002)