

Chapter 2

Fractional-Order Darwinian PSO

Abstract As presented in Chap. 1, Darwinian particle swarm optimization (DPSO) presented by Tillett et al. (Darwinian particle swarm optimization, 2005) is an evolutionary algorithm that extends the PSO using natural selection, or survival of the fittest, to enhance the ability to escape from local optima. Despite its superior performance when compared to its nonevolutionary counterpart, the DPSO also exhibits a key drawback: its computational complexity. This chapter proposes a method for controlling the convergence rate of the DPSO using fractional calculus (FC) concepts (Pires et al., Journal on Nonlinear Dynamics, 61(1–2), 295–301, 2010). The fractional-order optimization algorithm, denoted fractional-order Darwinian particle swarm optimization (FODPSO), is then tested using several well-known functions and the relationship between the fractional-order velocity and the convergence of the algorithm is observed.

Keywords FODPSO • Swarm intelligence • Optimization • Benchmark

2.1 Fractional Calculus

Fractional calculus (FC) has attracted the attention of several researchers (Sabatier et al. 2007), being applied in various scientific fields such as engineering, computational mathematics, and fluid mechanics, among others. FC can be considered as a generalization of integer-order calculus, thus accomplishing what integer-order calculus cannot. As a natural extension of the integer (i.e., classical) derivatives, fractional derivatives provide an excellent instrument for the description of memory and hereditary properties of processes.

The concept of the Grünwald–Letnikov fractional differential is presented by the following definition.

Definition 2.1 (Machado et al. 2010) Let Γ be the gamma function defined as

$$\Gamma(k) = (k - 1)!. \quad (2.1)$$

The signal $D^\alpha[x(t)]$ given by

$$D^\alpha[x(t)] = \lim_{h \rightarrow 0} \left[\frac{1}{h^\alpha} \sum_{k=0}^{+\infty} \frac{(-1)^k \Gamma(\alpha + 1) x(t - kh)}{\Gamma(k + 1) \Gamma(\alpha - k + 1)} \right], \quad (2.2)$$

is said to be the Grünwald–Letnikov fractional derivative of order α , $\alpha \in \mathbb{C}$, of a generic signal $x(t)$.

An important property revealed by Eq. (2.2) is that although an integer-order derivative just implies a finite series, the fractional-order derivative requires an infinite number of terms. Therefore, integer derivatives are “local” operators whereas fractional derivatives have, implicitly, a “memory” of all past events. However, the influence of past events decreases over time.

The formulation in (2.2) inspires a discrete time calculation presented by the following definition.

Definition 2.2 (Machado et al. 2010) The signal $D^\alpha[x[t]]$ given by

$$D^\alpha[x[t]] = \frac{1}{T^\alpha} \sum_{k=0}^r \frac{(-1)^k \Gamma[\alpha + 1] x[t - kT]}{\Gamma[k + 1] \Gamma[\alpha - k + 1]}, \quad (2.3)$$

where T is the sampling period and r is the truncation order, is the approximate discrete time Grünwald–Letnikov fractional difference of order α , $\alpha \in \mathbb{C}$, of the generic discrete signal $x[t]$.

The series presented in Eq. (2.3) can be implemented by a rational fraction expansion that leads to a superior compromise in what concerns the number of terms versus the quality of the approximation. Nevertheless, because this study focuses on the convergence of robots toward a given solution considering past events, the simple series approximation is adopted.

That being said, it is possible to extend an integer discrete difference, that is, classical discrete difference, to a fractional-order one, using the following definition.

Definition 2.3 (Ostalczyk 2009) The classical integer “direct” discrete difference of signal $x[t]$ is defined:

$$\Delta^d x[t] = \begin{cases} x[t], & d = 0 \\ x[t] - x[t - 1], & d = 1 \\ \Delta^{d-1} x[t] - \Delta^{d-1} x[t - 1], & d > 1 \end{cases} \quad (2.4)$$

where $d \in \mathbb{N}_0$ is the order of the integer discrete difference. Hence, one can extend the integer-order $\Delta^d x[t]$ assuming that the fractional discrete difference satisfies the following inequalities,

$$d - 1 < \alpha < d. \quad (2.5)$$

The features inherent to fractional calculus make this mathematical tool well suited to describe many phenomena, such as irreversibility and chaos, because of its inherent memory property. In this line of thought, the dynamic phenomena of particles' trajectories configure a case where fractional calculus tools fit adequately.

2.2 FODPSO

Considering the inertial influence of Eq. (1.1) as $w = 1$, for a specific swarm s , one would obtain:

$$v_n^s[t + 1] = v_n^s[t] + \sum_{i=1}^2 \rho_i r_i (\chi_{in}^s[t] - x_n^s[t]). \quad (2.6)$$

This expression can be rewritten:

$$v_n^s[t + 1] - v_n^s[t] = \sum_{i=1}^2 \rho_i r_i (\chi_{in}^s[t] - x_n^s[t]). \quad (2.7)$$

Hence, $v_n^s[t + 1] - v_n^s[t]$ corresponds to the discrete version of the fractional difference of order $\alpha = 1$, that is, the first-order integer difference $\Delta^d v_n^s[t + 1]$. Assuming $T = 1$ and based on Definition 2.2, yields the equation:

$$D^\alpha [v_n^s[t + 1]] = \sum_{i=1}^2 \rho_i r_i (\chi_{in}^s[t] - x_n^s[t]). \quad (2.8)$$

Based on the FC concept and Definition 2.3, the order of the velocity derivative can be generalized to a real number $0 < \alpha < 1$, thus leading to a smoother variation and a longer memory effect. Therefore, considering the discrete-time fractional differential presented in Definition 2.2, one can rewrite Eq. (2.8) as

$$\begin{aligned} v_n^s[t + 1] = & - \sum_{k=1}^r \frac{(-1)^k \Gamma[\alpha + 1] v_n^s[t + 1 - kT]}{\Gamma[k + 1] \Gamma[\alpha - k + 1]} \\ & + \sum_{i=1}^2 \rho_i r_i (\chi_{in}^s[t] - x_n^s[t]). \end{aligned} \quad (2.9)$$

The DPSO is, therefore, a particular case of the FODPSO for $\alpha = 1$ (without “memory”).

2.2.1 Benefits

The FODPSO is, in simple terms, the same as having multiple PSOs, wherein particles strive to find the best solution for their own “survival”, with the perk of intrinsically having a memory of past decisions. This new architecture handles the first drawback pointed out for the traditional PSO: the premature convergence of a swarm. The FODPSO, as does the traditional DPSO (Tillett et al. 2005), discards swarms that prematurely converge toward a solution that may, or may not, be optimal. At the same time, it fosters the creation of new swarms formed by particles that may “genetically” share some of the knowledge already retrieved by other particles. Additionally to this, each FODPSO particle is considerably “smarter” than PSO and DPSO particles due to the fractional-order extension that improves the balance between exploration and exploitation (see Sect. 2.3). This allows running the FODPSO algorithm with a smaller population when compared to the DPSO algorithm, thus reducing the computational complexity, and still expecting the same end result. This book repeatedly compares the FODPSO with alternatives, including the PSO and DPSO, and clearly depicts its superiority in every aspect.

2.2.2 Drawbacks

Compared to the other PSO-based alternatives, from which the traditional PSO and the DPSO algorithms are used as references, the FODPSO presents two drawbacks: (i) its memory complexity and (ii) the addition of a new coefficient α .

As opposed to most PSO-based approaches that only require memorizing the previous iteration, as one can observe from Eq. (2.9), computing a new velocity at time $t + 1$ requires memorizing the previous r iterations. Therefore, one needs to ensure a proper r , being large enough to ensure an improved convergence of particles toward the solution when compared to the alternatives, while at the same time small enough not to increase the cost of the algorithm significantly.

Preliminary experimental tests on the algorithm presented similar results for ≥ 4 (Couceiro et al. 2012). Furthermore, the computational requirements increase linearly with r ; that is, the FODPSO presents an $\mathcal{O}(r)$ memory complexity. Hence, using only the first $r = 4$ terms of differential derivative given by (2.3), Eq. (2.9) can be rewritten as (2.10):

$$\begin{aligned}
v_n^s[t+1] = & \alpha v_n^s[t] + \frac{1}{2} \alpha (1 - \alpha) v_n^s[t-1] \\
& + \frac{1}{6} \alpha (1 - \alpha) (2 - \alpha) v_n^s[t-2] \\
& + \frac{1}{24} \alpha (1 - \alpha) (2 - \alpha) (3 - \alpha) v_n^s[t-3] \\
& + \sum_{i=1}^2 \rho_i r_i (\chi_{in}^s[t] - x_n^s[t])
\end{aligned} \tag{2.10}$$

Although this new equation incorporates the concept of FC, the difficulty of understanding the influence inherent to the fractional coefficient α still remains: what should be the most adequate value for α ?

As described in Yasuda et al. (2008) and Wakasa et al. (2010), a swarm behavior can be divided into two activities: (i) exploitation and (ii) exploration. The first one is related to the convergence of the algorithm, thus allowing a good short-term performance. However, if the exploitation level is too high, then the algorithm may be stuck on local solutions. The second one is related to the diversification of the algorithm, which allows exploring new solutions thus improving the long-term performance. However, if the exploration level is too high, then the algorithm may take too much time to find the global solution. As first presented by Shi and Eberhart (2001), the trade-off between exploitation and exploration in the classical PSO has been commonly handled by adjusting the inertia weight. A large inertia weight improves exploration activity whereas exploitation is improved using a small inertia weight. Because the FODPSO presents a FC strategy to control the convergence of particles, the coefficient α needs to be defined in order to provide a high level of exploration while ensuring the global solution of the mission. Therefore, the FODPSO is experimentally evaluated in the next section using Eq. (2.10) for all particles in all swarms.

2.3 Benchmarking Functions

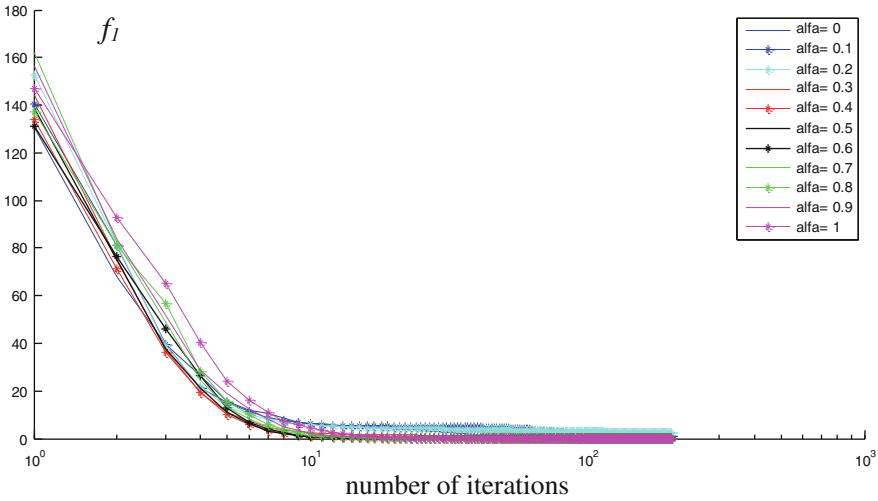
This section presents experimental results of the proposed FODPSO. In order to compare this approach with the Pires et al. approach (Pires et al. 2010), the same test functions and parameters are considered as depicted in Table 2.1. Table 2.1 also shows the specific parameters of the FODPSO algorithm.

The median of the fitness evolution of the best global particle is taken as the system output: for each value in the set $\alpha = \{0, 0.1, \dots, 1\}$. In Figs. 2.1, 2.2, 2.3, 2.4 and 2.5, the results can be seen for the adopted optimization functions $f_j, j = \{1, \dots, 5\}$.

Experimental results show that the convergence of the algorithm depends upon the fractional order α . However, contrary to the FOPSO presented in Pires et al. (2010), the Darwinian algorithm easily avoids being stuck in local solutions

Table 2.1 Parameters of the algorithm and optimization functions

	Min	Initial	Max
Number of simulations	–	201	–
Number of iterations	–	200	–
Coefficients $\rho_1 = \rho_2$	–	0.8	–
Swarm population	3	4	5
Number of swarms	1	2	3
Stagnancy threshold	–	10	–
Optimization functions f_j (Van Den Bergh and Engelbrecht 2006)	Bohachevsky 1		
	Colville		
	Drop wave		
	Easom		
	Rastrigin		

**Fig. 2.1** Evolution of the Bohachevsky 1 function changing α

independently of the value of α (because it is a particularity of the traditional DPSO). Moreover, one can observe that, in most situations, a faster optimization convergence is obtained for a fractional coefficient α in the range $[0.5, 0.8]$. Therefore, to evaluate the FODPSO further, let us then systematically adjust the fractional coefficient α between 0.5 and 0.8, according to the expression:

$$\alpha(t) = 0.8 - 0.3 \frac{t}{200}. \quad (2.11)$$

Once again, the median of the fitness evolution of the best global particle is taken as the system output. In Fig. 2.6, the results can be seen for the adopted optimization

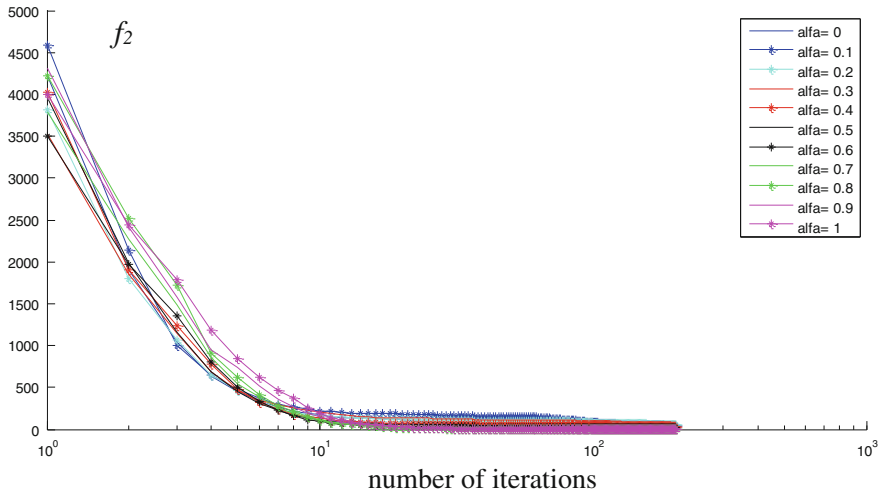


Fig. 2.2 Evolution of the Colville function changing α

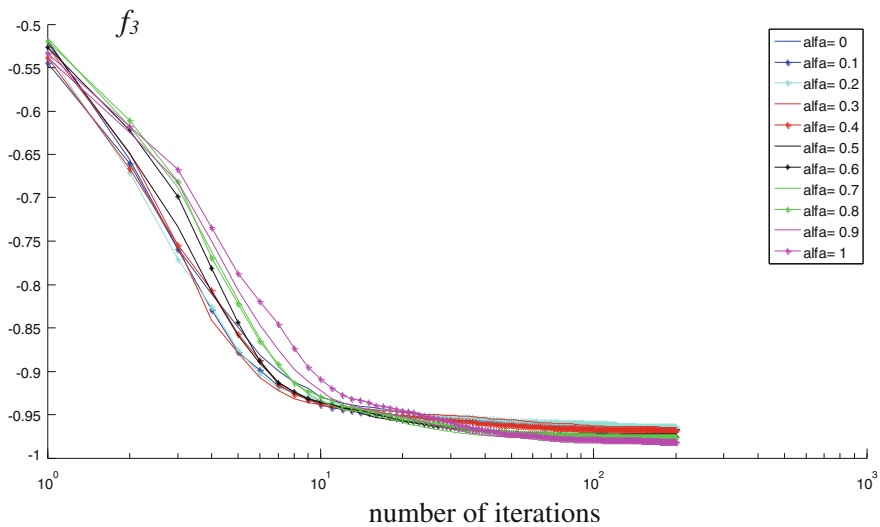


Fig. 2.3 Evolution of the drop wave function changing α

functions $f_j, j = \{1, \dots, 5\}$, while comparing the FODPSO with the FOPSO proposed by Pires et al. (2010) using Eq. (2.11). Observing Fig. 2.6, one can conclude that, despite both FOPSO and FODPSO revealing a similar behavior, the combination of FC and Darwin's principles contributes to an improved convergence dynamics.

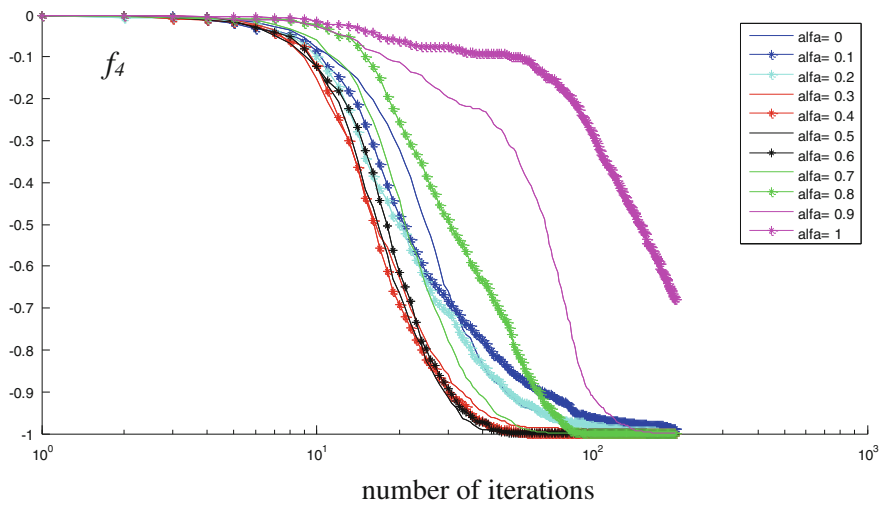


Fig. 2.4 Evolution of the Easom function changing α

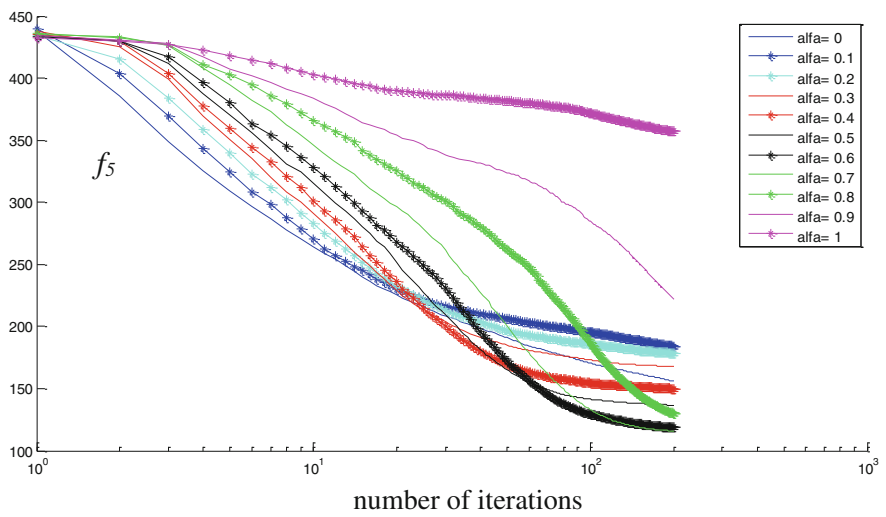


Fig. 2.5 Evolution of the Rastrigin function changing α

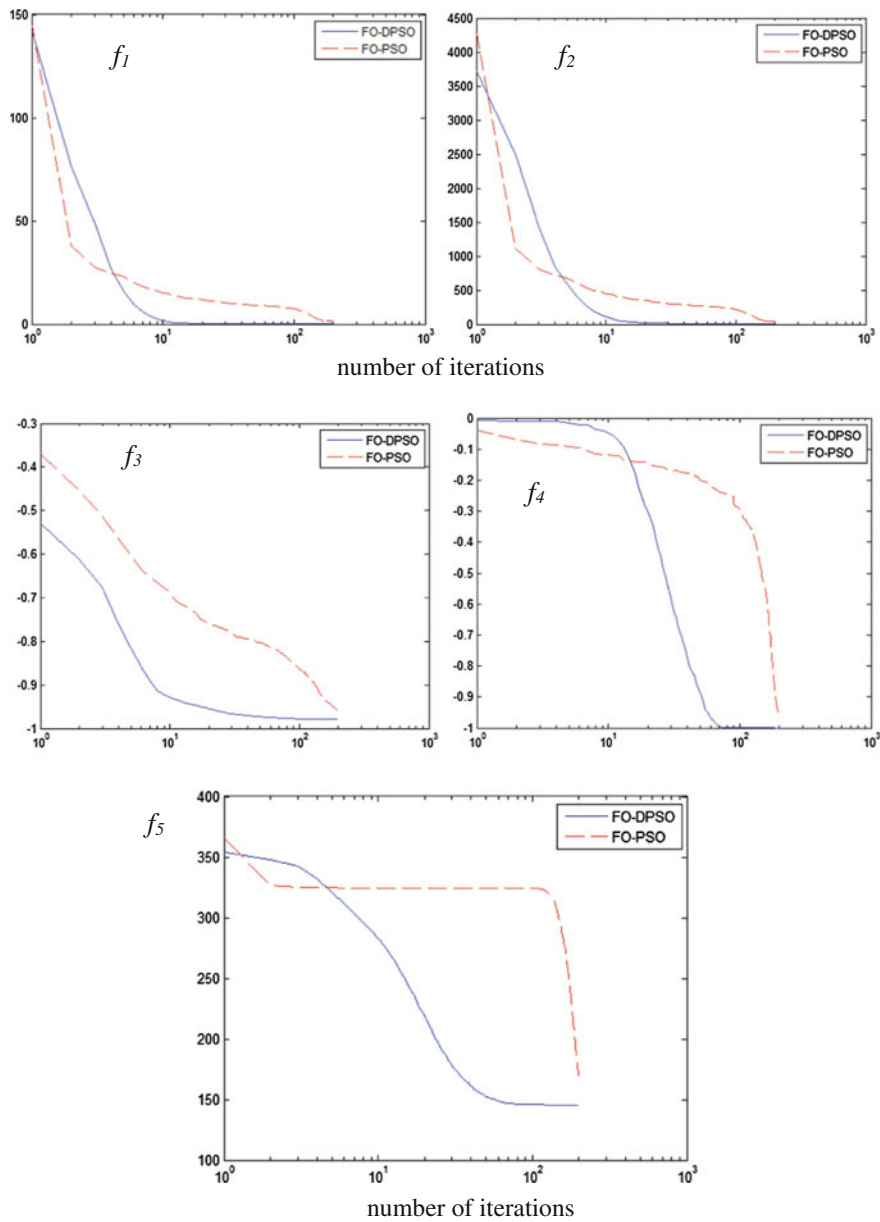


Fig. 2.6 Evolution of the fitness function, with variable α for FOPSO and FODPSO

2.4 Summary

The search for an algorithm capable of dealing with most optimization problems without being very time-consuming and computationally demanding has been a subject of research in several scientific areas such as control engineering and applied mathematics. Fractional calculus has appeared as a tool to enhance the performance of conventional mathematical methods.

This chapter presented a new optimization algorithm based on the Darwinian particle swarm optimization (DPSO) using the concept of the fractional derivative to control the convergence rate.

Experimental results show that, although the speed of convergence of the fractional order DPSO (FODPSO) depends on the fractional order α , the herein presented algorithm outperforms the traditional DPSO and PSO, as well as the FOPSO previously presented in the literature.

References

- Couceiro, M. S., Martins, F. M., Rocha, R. P., & Ferreira, N. M. (2012). Introducing the fractional order robotic Darwinian PSO. *Proceedings of the 9th International Conference on Mathematical Problems in Engineering, Aerospace and Sciences (ICNPAA2012)* (pp. 242–251). Vienna, Austria.
- Machado, J. T., Silva, M. F., Barbosa, R. S., Jesus, I. S., Reis, C. M., Marcos, M. G., et al. (2010). Some applications of fractional calculus in engineering, Hindawi Publishing Corporation *Mathematical Problems in Engineering*, 1(34), 174–183.
- Ostalczyk, P. W. (2009). A note on the Grünwald-Letnikov fractional-order backward-difference. *Physica Scripta*, 136, 014036.
- Pires, E. J., Machado, J. A., Cunha, P. B., & Mendes, L. (2010). Particle swarm optimization with fractional-order velocity. *Journal on Nonlinear Dynamics*, 61(1–2), 295–301.
- Sabatier, J., Agrawal, O. P., & Machado, J. A. (2007). *Advances in fractional calculus—theoretical developments and applications in physics and engineering*. Berlin: Springer.
- Shi, Y., & Eberhart, R. (2001). Fuzzy adaptive particle swarm optimization. *Proceedings of the IEEE Congress on Evolutionary Computation*, 1, 101–106.
- Tillett, J., Rao, T. M., Sahin, F., Rao, R., & Brockport, S. (2005). Darwinian particle swarm optimization. In B. Prasad (Ed.), *Proceedings of the 2nd Indian International Conference on Artificial Intelligence* (pp. 1474–1487). Pune, India.
- Van Den Bergh, F., & Engelbrecht, A. P. (2006). *A study of particle swarm optimization particle trajectories*, 176(8), 937–971.
- Wakasa, Y., Tanaka, K., & Nishimura, Y. (2010). Control-theoretic analysis of exploitation and exploration of the PSO algorithm. *IEEE International Symposium on Computer-Aided Control System Design, IEEE Multi-Conference on Systems and Control* (pp. 1807–1812). Yokohama, Japan.
- Yasuda, K., Iwasaki, N., Ueno, G., & Aiyoshi, E. (2008). Particle swarm optimization: A numerical stability analysis and parameter adjustment based on swarm activity. *IEEE Transactions on Electrical and Electronic Engineering, Wiley InterScience*, 3, 642–659.

Fractional Order Darwinian Particle Swarm Optimization
Applications and Evaluation of an Evolutionary
Algorithm

Couceiro, M.; Ghamisi, P.

2016, X, 75 p. 27 illus., 24 illus. in color., Softcover

ISBN: 978-3-319-19634-3