

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

Our authors are among the

154

Countries delivered to

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



A Particle Swarm Optimization technique used for the improvement of analogue circuit performances

Mourad Fakhfakh¹, Yann Cooren², Mourad Loulou¹ and Patrick Siarry²
¹University of Sfax, ²University of Paris 12
¹Tunisia, ²France

1. Introduction

The importance of the analogue part in integrated electronic systems cannot be overstressed. Despite its eminence, and unlike the digital design, the analogue design has not so far been automated to a great extent, mainly due to its towering complexity (Dastidar et al., 2005). Analogue sizing is a very complicated, iterative and boring process whose automation is attracting great attention (Medeiro et al., 1994). The analogue design and sizing process remains characterized by a mixture of experience and intuition of skilled designers (Tlelo-Cuautle & Duarte-Villaseñor, 2008). As a matter of fact, optimal design of analogue components is over and over again a bottleneck in the design flow.

Optimizing the sizes of the analogue components automatically is an important issue towards ability of rapidly designing true high performance circuits (Toumazou & Lidgley, 1993; Conn et al., 1996).

Common approaches are generally either fixed topology ones or/and statistical-based techniques. They generally start with finding a “good” DC quiescent point, which is provided by the skilled analogue designer. After that a simulation-based tuning procedure takes place. However these statistic-based approaches are time consuming and do not guarantee the convergence towards the global optimum solution (Talbi, 2002).

Some mathematical heuristics were also used, such as Local Search (Aarts & Lenstra, 2003), Simulated Annealing (Kirkpatrick et al., 1983; Siarry(a) et al., 1997), Tabu Search (Glover, 1989; Glover, 1990), Genetic Algorithms (Grimbleby, 2000; Dréo et al., 2006), etc. However these techniques do not offer general solution strategies that can be applied to problem formulations where different types of variables, objectives and constraint functions are used. In addition, their efficiency is also highly dependent on the algorithm parameters, the dimension of the solution space, the convexity of the solution space, and the number of variables.

Actually, most of the circuit design optimization problems simultaneously require different types of variables, objective and constraint functions in their formulation. Hence, the abovementioned optimization procedures are generally not adequate or not flexible enough.

In order to overcome these drawbacks, a new set of nature inspired heuristic optimization algorithms were proposed. The thought process behind these algorithms is inspired from

the collective behaviour of decentralized, self-organized systems. It is known as Swarm Intelligence (SI) (Bonabeau et al. 1999). SI systems are typically made up of a population of simple agents (or "particles") interacting locally with each other and with their environment. These particles obey to very simple rules, and although there is no centralized control structure dictating how each particle should behave, local interactions between them lead to the emergence of complex global behaviour. Most famous such SIs are Ant Colony Optimization (ACO) (Dorigo et al., 1999), Stochastic Diffusion Search (SDS) (Bishop, 1989) and Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995; Clerc, 2006).

PSO, in its current form, has been in existence for almost a decade, which is a relatively short period when compared to some of the well known natural computing paradigms, such as evolutionary computation. PSO has gained widespread demand amongst researchers and has been shown to offer good performance in an assortment of application domains (Banks et al., 2007).

In this chapter, we focus on the use of PSO technique for the optimal design of analogue circuits. The practical applicability and suitability of PSO to optimize performances of such multi-objective problems are highlighted. An example of optimizing performances of a second generation MOS current conveyor (CCII) is presented. The used PSO algorithm is detailed and Spice simulation results, performed using the 'optimal' sizing of transistors forming the CCII and bias current, are presented. Reached performances are discussed and compared to others presented in some published works, but obtained using classical approaches.

2. The Sizing Problem

The process of designing an analogue circuit mainly consists of the following steps (Medeiro et al., 1994) :

- the topology choice: a suitable schematic has to be selected,
- the sizing task: the chosen schematic must be dimensioned to comply with the required specifications,
- The generation of the layout.

Among these major steps, we focus on the second one, i.e. the optimal sizing of analogue circuits.

Actually, analogue sizing is a constructive procedure that aims at mapping the circuit specifications (objectives and constraints on performances) into the design parameter values. In other words, the performance metrics of the circuit, such as gain, noise figure, input impedance, occupied area, etc. have to be formulated in terms of the design variables (Tulunay & Balkir, 2004).

In a generic circuit, the optimization problem consists of finding optimal values of the design parameters. These variables form a vector $\vec{X}^T = \{x_1, x_2, \dots, x_N\}$ belonging to an N-dimensional design space. This set includes transistor geometric dimensions and passive component values, if any. Hence, performances and objectives involved in the design objectives are expressed as functions of X.

These performances may belong to the set of constraints ($\vec{g}(\vec{X})$) and/or to the set of objectives ($\vec{f}(\vec{X})$). Thus, a general optimization problem can be formulated as follows:

$$\begin{aligned}
 \text{minimize} & \quad : f_i(\vec{X}), i \in [1, k] \\
 \text{such that:} & \quad : g_j(\vec{X}) \leq 0, j \in [1, l] \\
 & \quad h_m(\vec{X}) = 0, m \in [1, p] \\
 & \quad x_{Li} \leq x_i \leq x_{Ui}, i \in [1, N]
 \end{aligned} \tag{1}$$

k, l and p denote the numbers of objectives, inequality constraints and equality constraints, respectively. \vec{x}_L and \vec{x}_U are lower and upper boundaries vectors of the parameters.

The goal of optimization is usually to minimize an objective function; the problem for maximizing $\vec{f}(\vec{x})$ can be transformed into minimizing $-\vec{f}(\vec{x})$. This goal is reached when the variables are located in the set of optimal solutions.

For instance, a basic two-stage operational amplifier has around 10 parameters, which include the widths and lengths of all transistors values which have to be set. The goal is to achieve around 10 specifications, such as gain, bandwidth, noise, offset, settling time, slew rate, consumed power, occupied area, CMRR (common-mode rejection ratio) and PSRR (power supply rejection ratio). Besides, a set of DC equations and constraints, such as transistors' saturation conditions, have to be satisfied (Gray & Meyer, 1982).

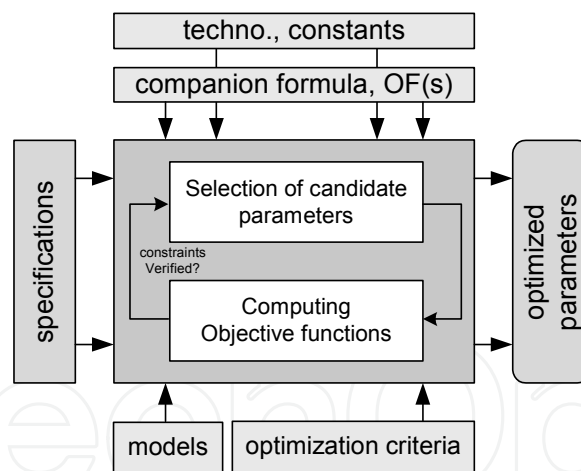


Figure 1. Pictorial view of a design optimization approach

The pictorial flow diagram depicted in Fig. 1 summarizes main steps of the sizing approach. As it was introduced in section 1, there exist many papers and books dealing with mathematic optimization methods and studying in particular their convergence properties (see for example (Talbi, 2002; Dréo et al., 2006; Siarry(b) et al., 2007)).

These optimizing methods can be classified into two categories: deterministic methods and stochastic methods, known as heuristics.

Deterministic methods, such as Simplex (Nelder & Mead, 1965), Branch and Bound (Doig, 1960), Goal Programming (Scniederjans, 1995), Dynamic Programming (Bellman, 2003)... are effective only for small size problems. They are not efficient when dealing with NP-hard

and multi-criteria problems. In addition, it has been proven that these optimization techniques impose several limitations due to their inherent solution mechanisms and their tight dependence on the algorithm parameters. Besides they rely on the type of objective, the type of constraint functions, the number of variables and the size and the structure of the solution space. Moreover they do not offer general solution strategies.

Most of the optimization problems require different types of variables, objective and constraint functions simultaneously in their formulation. Therefore, classic optimization procedures are generally not adequate.

Heuristics are necessary to solve big size problems and/or with many criteria (Basseur et al., 2006). They can be 'easily' modified and adapted to suit specific problem requirements. Even though they don't guarantee to find in an exact way the optimal solution(s), they give 'good' approximation of it (them) within an acceptable computing time (Chan & Tiwari, 2007). Heuristics can be divided into two classes: on the one hand, there are algorithms which are specific to a given problem and, on the other hand, there are generic algorithms, i.e. metaheuristics. Metaheuristics are classified into two categories: local search techniques, such as Simulated Annealing, Tabu Search ... and global search ones, like Evolutionary techniques, Swarm Intelligence techniques ...

ACO and PSO are swarm intelligence techniques. They are inspired from nature and were proposed by researchers to overcome drawbacks of the aforementioned methods. In the following, we focus on the use of PSO technique for the optimal design of analogue circuits.

3. Overview of Particle Swarm Optimization

The particle swarm optimization was formulated by (Kennedy & Eberhart, 1995). The cogitated process behind the PSO algorithm was inspired by the optimal swarm behaviour of animals such, as birds, fishes and bees.

PSO technique encompasses three main features:

- It is a SI technique; it mimics some animal's problem solution abilities,
- It is based on a simple concept. Hence, the algorithm is neither time consumer nor memory absorber,
- It was originally developed for continuous nonlinear optimization problems. As a matter of fact, it can be easily expanded to discrete problems.

PSO is a stochastic global optimization method. Like in Genetic Algorithms (GA), PSO exploits a population of potential candidate solutions to investigate the feasible search space. However, in contrast to GA, in PSO no operators inspired by natural evolution are applied to extract a new generation of feasible solutions. As a substitute of mutation, PSO relies on the exchange of information between individuals (particles) of the population (swarm).

During the search for the promising regions of the landscape, and in order to tune its trajectory, each particle adjusts its velocity and its position according to its own experience, as well as the experience of the members of its social neighbourhood. Actually, each particle remembers its best position, and is informed of the best position reached by the swarm, in the global version of the algorithm, or by the particle's neighbourhood, in the local version of the algorithm. Thus, during the search process, a global sharing of information takes place and each particle's experience is thus enriched thanks to its discoveries and those of all the other particles. Fig. 2 illustrates this principle.

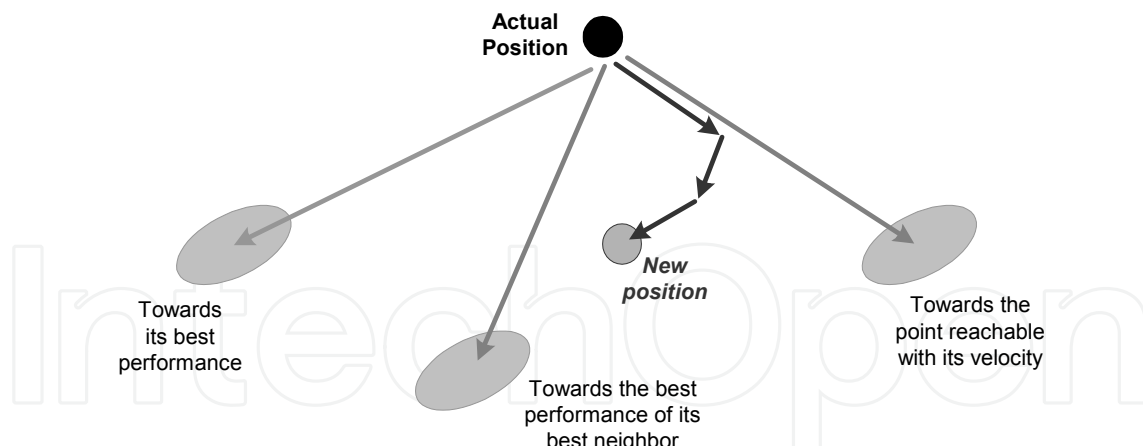


Figure 2. Principle of the movement of a particle

In an N-dimensional search space, the position and the velocity of the i^{th} particle can be represented as $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,N}]$ and $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,N}]$ respectively. Each particle has its own best location $P_i = [p_{i,1}, p_{i,2}, \dots, p_{i,N}]$, which corresponds to the best location reached by the i^{th} particle at time t . The global best location is named $g = [g_1, g_2, \dots, g_N]$, which represents the best location reached by the entire swarm. From time t to time $t+1$, each velocity is updated using the following equation:

$$v_{i,j}(t+1) = \underbrace{w}_{\text{inertia}} v_{i,j}(t) + \underbrace{c_1 r_1}_{\text{Personal Influence}} (p_{i,j} - v_{i,j}(t)) + \underbrace{c_2 r_2}_{\text{Social Influence}} (g_i - v_{i,j}(t)) \quad (2)$$

where w is a constant known as inertia factor, it controls the impact of the previous velocity on the current one, so it ensures the diversity of the swarm, which is the main means to avoid the stagnation of particles at local optima. c_1 and c_2 are constants called acceleration coefficients; c_1 controls the attitude of the particle of searching around its best location and c_2 controls the influence of the swarm on the particle's behaviour. r_1 and r_2 are two independent random numbers uniformly distributed in $[0,1]$.

The computation of the position at time $t+1$ is derived from expression (2) using:

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (3)$$

It is important to put the stress on the fact that the PSO algorithm can be used for both mono-objective and multi-objective optimization problems.

The driving idea behind the multi-objective version of PSO algorithm (MO-PSO) consists of the use of an archive, in which each particle deposits its 'flight' experience at each running cycle. The aim of the archive is to store all the non-dominated solutions found during the optimization process. At the end of the execution, all the positions stored in the archive give us an approximation of the theoretical Pareto Front. Fig. 3 illustrates the flowchart of the MO-PSO algorithm. Two points are to be highlighted: the first one is that in order to avoid excessive growing of the storing memory, its size is fixed according to a crowding rule (Cooren et al., 2007). The second point is that computed optimal solutions' inaccuracy crawls in due to the inaccuracy of the formulated equations.

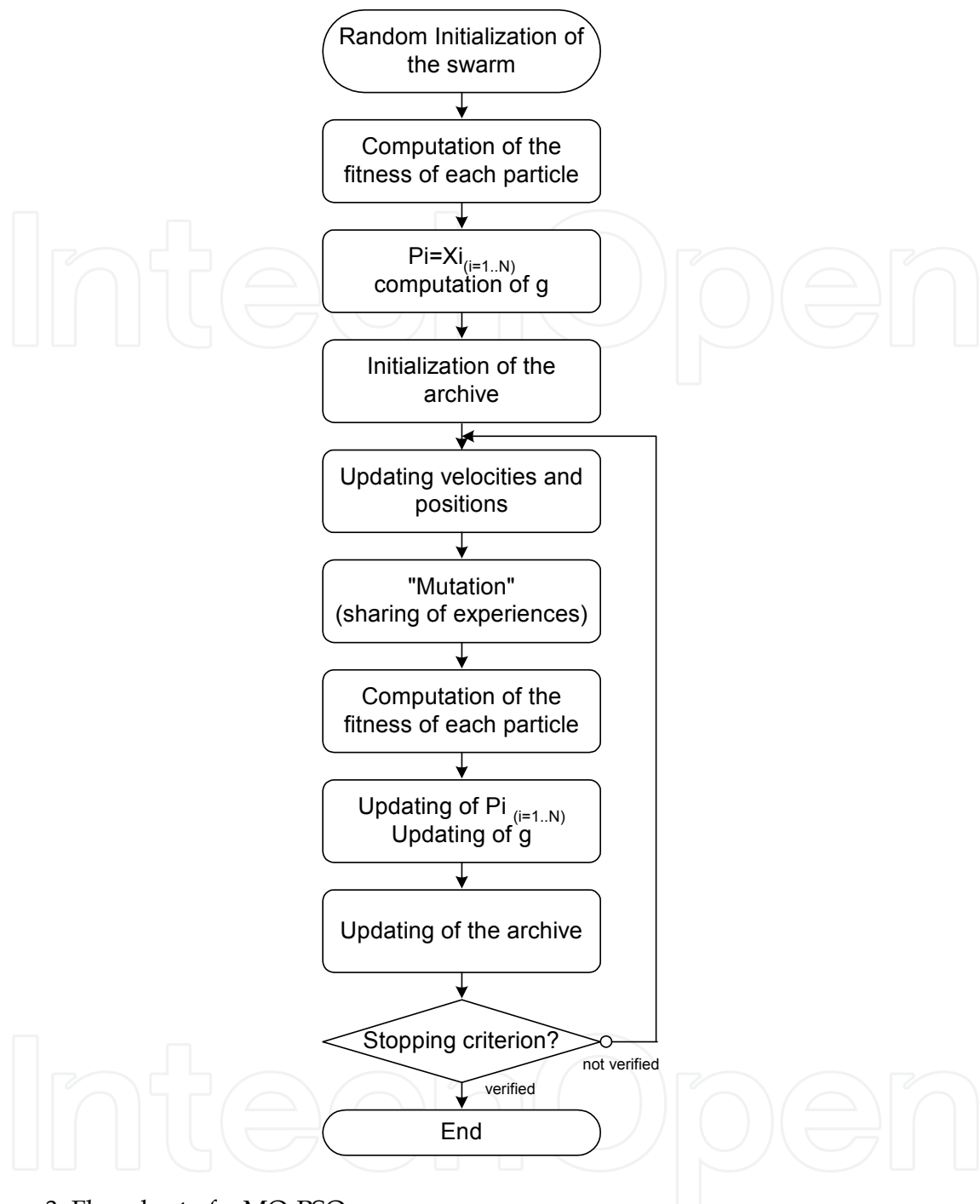


Figure 3. Flowchart of a MO-PSO

In the following section we give an application example dealing with optimizing performances of an analogue circuit, i.e. optimizing the sizing of a MOS inverted current conveyor in order to maximize/minimize performance functions, while satisfying imposed and inherent constraints. The problem consists of generating the trade off surface (Pareto front¹) linking two conflicting performances of the CCII, namely the high cut-off current frequency and the parasitic X-port input resistance.

¹ Definition of Pareto optimality is given in Appendix.

4. An Application Example

The problem consists of optimizing performances of a second generation current conveyor (CCII) (Sedra & Smith, 1970) regarding to its main influencing performances. The aim consists of maximizing the conveyor high current cut-off frequency and minimizing its parasitic X-port resistance (Cooren et al., 2007).

In the VLSI realm, circuits are classified according to their operation modes: voltage mode circuits or current mode circuits. Voltage mode circuits suffer from low bandwidths arising due to the stray and circuit capacitances and are not suitable for high frequency applications (Rajput & Jamuar, 2007).

In contrary, current mode circuits enable the design of circuits that can operate over wide dynamic ranges. Among the set of current mode circuits, the current conveyor (CC) (Smith & Sedra, 1968; Sedra & Smith, 1970) is the most popular one.

The Current Conveyor (CC) is a three (or more) terminal active block. Its conventional representation is shown in Fig. 4a. Fig. 4b shows the equivalent nullator/norator representation (Schmid, 2000) which reproduces the ideal behaviour of the CC. Fig. 4.c shows a CCII with its parasitic components (Ferry et al. 2002).

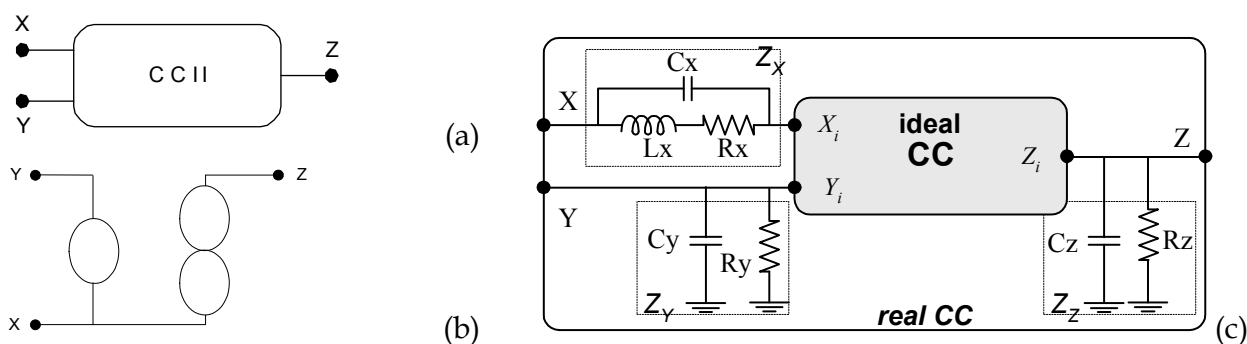


Figure 4. (a) General representation of current conveyor, (b) the nullor equivalency: ideal CC, (c) parasitic components: real CC

Relations between voltage and current terminals are given by the following matrix relation (Toumazou & Lidgey, 1993):

$$\begin{bmatrix} I_Y \\ V_X \\ I_Z \end{bmatrix} = \begin{bmatrix} \frac{1}{(C_Y // R_Y)} & \alpha & 0 \\ \gamma & R_X & 0 \\ 0 & \beta & \frac{1}{(C_Z // R_Z)} \end{bmatrix} \begin{bmatrix} V_Y \\ I_X \\ V_Z \end{bmatrix} \quad (4)$$

For the above matrix representation, α specifies the kind of the conveyor. Indeed, for $\alpha = 1$, the circuit is considered as a first generation current conveyor (CCI). Whereas when $\alpha = 0$, it is called a second generation current conveyor (CCII). β characterizes the current transfer from X to Z ports. For $\beta = +1$, the circuit is classified as a positive transfer conveyor. It is considered as a negative transfer one when $\beta = -1$. $\gamma = \pm 1$: When $\gamma = -1$ the CC is said an inverted CC, and a direct CC, otherwise.

Accordingly, the CCII ensures two functionalities between its terminals:

- A Current follower/Current mirror between terminals X and Z.
- A Voltage follower/Voltage mirror between terminals X and Y.

In order to get ideal transfers, CCII are commonly characterized by low impedance on terminal X and high impedance on terminals Y and Z.

In this application we deal with optimizing performances of an inverted positive second generation current conveyor (CCII+) (Sedra & Smith, 1970; Cooren et al., 2007) regarding to its main influencing performances. The aim consists of determining the optimal Pareto circuit's variables, i.e. widths and lengths of each MOS transistor, and the bias current I_0 , that maximizes the conveyor high current cut-off frequency and minimizes its parasitic X-port resistance (R_X) (Bensalem et al., 2006; Fakhfakh et al. 2007). Fig. 5 illustrates the CCII+'s MOS transistor level schema.

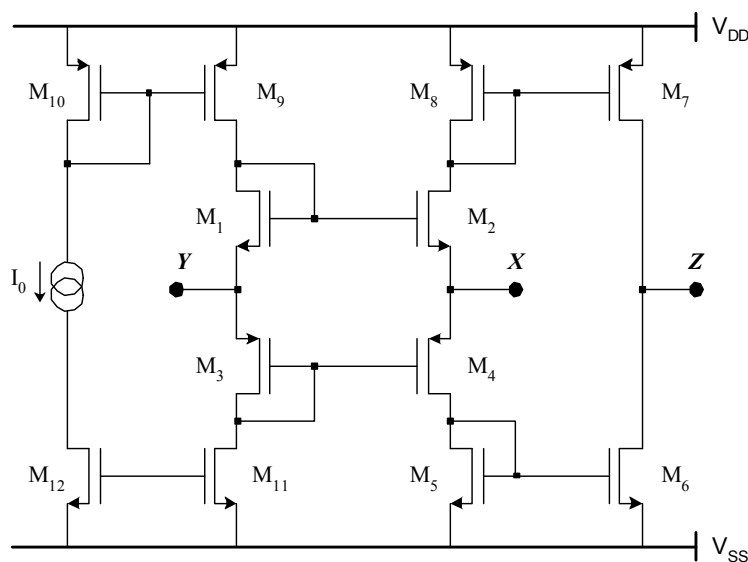


Figure 5. The second generation CMOS current conveyor

Constraints:

- Transistor saturation conditions: all the CCII transistors must operate in the saturation mode. Saturation constraints of each MOSFET were determined. For instance, expression (5) gives constraints on M2 and M8 transistors:

$$\sqrt{\frac{I_0}{K_P W_P / L_P}} \leq \frac{V_{DD}}{2} - |V_{TP}| - \sqrt{\frac{I_0}{K_N W_N / L_N}} \quad (5)$$

where I_0 is the bias current, $W_{(N,P)}/L_{(N,P)}$ is the aspect ratio of the corresponding MOS transistor. $K_{(N,P)}$ and V_{TP} are technology parameters. V_{DD} is the DC voltage power supply.

Objective functions:

In order to present simplified expressions of the objective functions, all NMOS transistors were supposed to have the same size. Ditto for the PMOS transistors.

- R_X : the value of the X-port input parasitic resistance has to be minimized,
- f_{chi} : the high current cut-off frequency has to be maximized.

Symbolic expressions of the objective functions are not given due to their large number of terms.

PSO algorithm was programmed using C++ software. Table 1 gives the algorithm parameters.

Fig. 6 shows Pareto fronts (R_X vs. f_{ci}) and optimal variables (W_P vs. W_N) corresponding to the bias current $I_0=50\mu A$ (6.a, 6.b), $100\mu A$ (6.c, 6.d), $150\mu A$ (6.e, 6.f), $200\mu A$ (6.g, 6.h), $250\mu A$ (6.i, 6.j) and $300\mu A$ (6.k, 6.l). Where values of L_N , L_P , W_N and W_P are given in μm , I_0 is in μA , R_X in ohms and $f_{ci(min, Max)}$ in GHz.

In Fig. 6 clearly appears the high interest of the Pareto front. Indeed, amongst the set of the non-dominated solutions, the designer can choose, always with respect to imposed specifications, its best solution since he can add some other criterion choice, such as Y-port and/or Z-port impedance values, high voltage cut-off frequency, etc.

Fig. 7 shows Spice simulation results performed for both points corresponding to the edge of the Pareto front, for $I_0=100\mu A$, where $R_{Xmin}=493$ ohms, $R_{XMax}=787$ ohms, $f_{ciMin}=0.165$ GHz and $f_{ciMax}=1.696$ GHz.

Swarm size	Number of iterations	w	c_1	c_2
20	1000	0.4	1	1

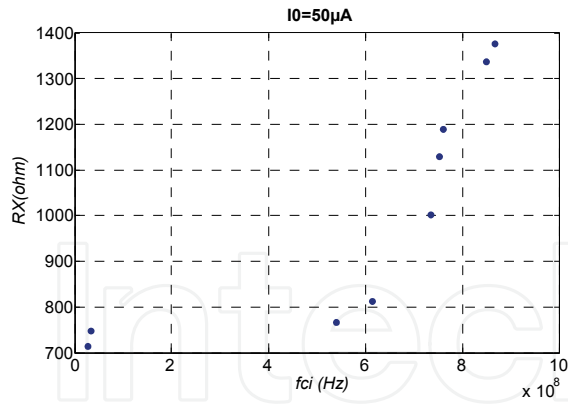
Table 1. The PSO algorithm parameters

Technology	CMOS AMS 0.35 μm
Power voltage supply	$V_{SS}=-2.5V$, $V_{DD}=2.5V$

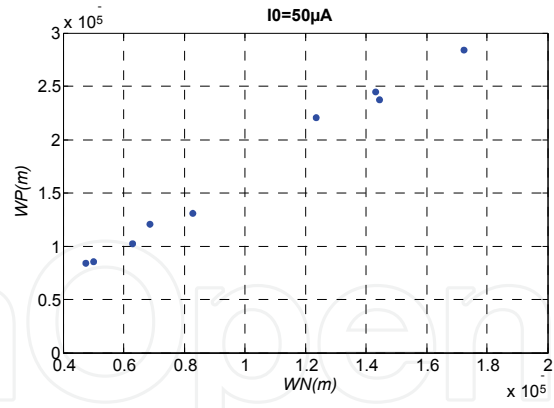
Table 2. SPICE simulation conditions

I_0	W_N	L_N	W_P	L_P	W_N	L_N	W_P	L_P
	R_{Xmin}		f_{ciMin}		R_{XMax}		f_{ciMax}	
50	17.21	0.90	28.40	0.50	4.74	0.87	8.40	0.53
	714		0.027		1376		0.866	
100	20.07	0.57	30.00	0.35	7.28	0.55	12.60	0.35
	382		0.059		633		1.802	
150	17.65	0.6	28.53	0.35	10.67	0.59	17.77	0.36
	336		0.078		435		1.721	
200	17.51	0.53	29.55	0.35	12.43	0.53	20.32	0.35
	285		0.090		338		2.017	
250	18.60	0.54	30.00	0.35	15.78	0.55	24.92	0.35
	249		0.097		272		1.940	
300	19.17	0.55	29.81	0.35	17.96	0.54	29.16	0.35
	224		0.107		230		2.042	

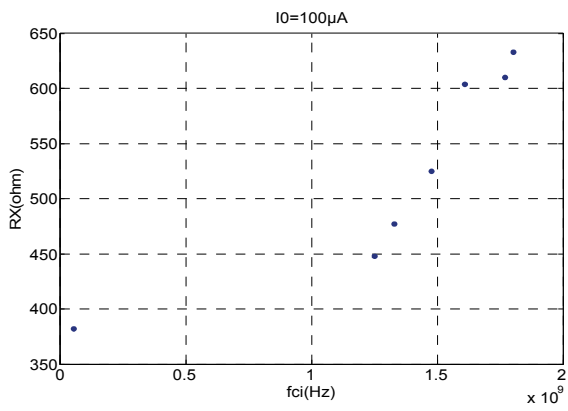
Table 3. Pareto trade-off surfaces' boundaries corresponding to some selected results



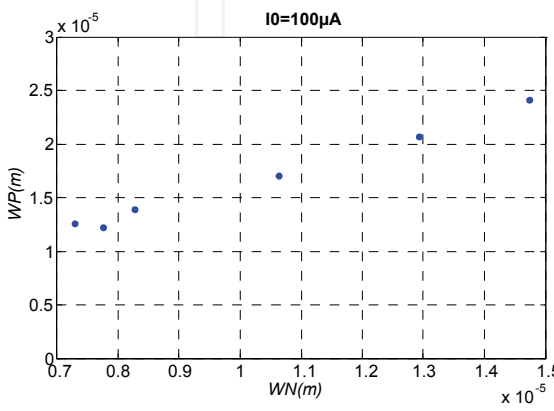
(a)



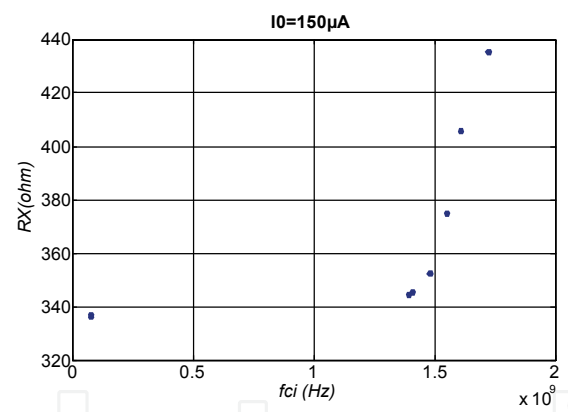
(b)



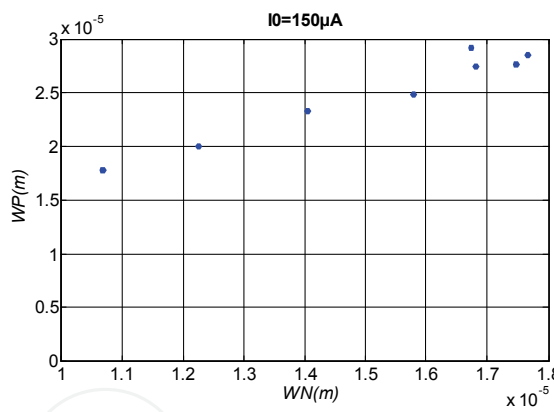
(c)



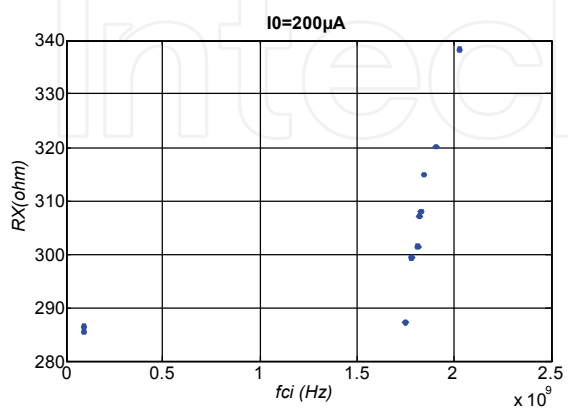
(d)



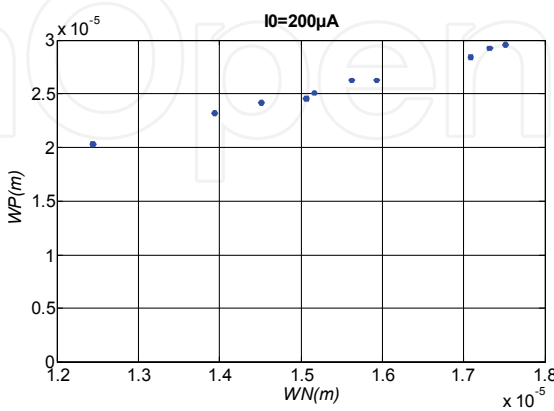
(e)



(f)



(g)



(h)

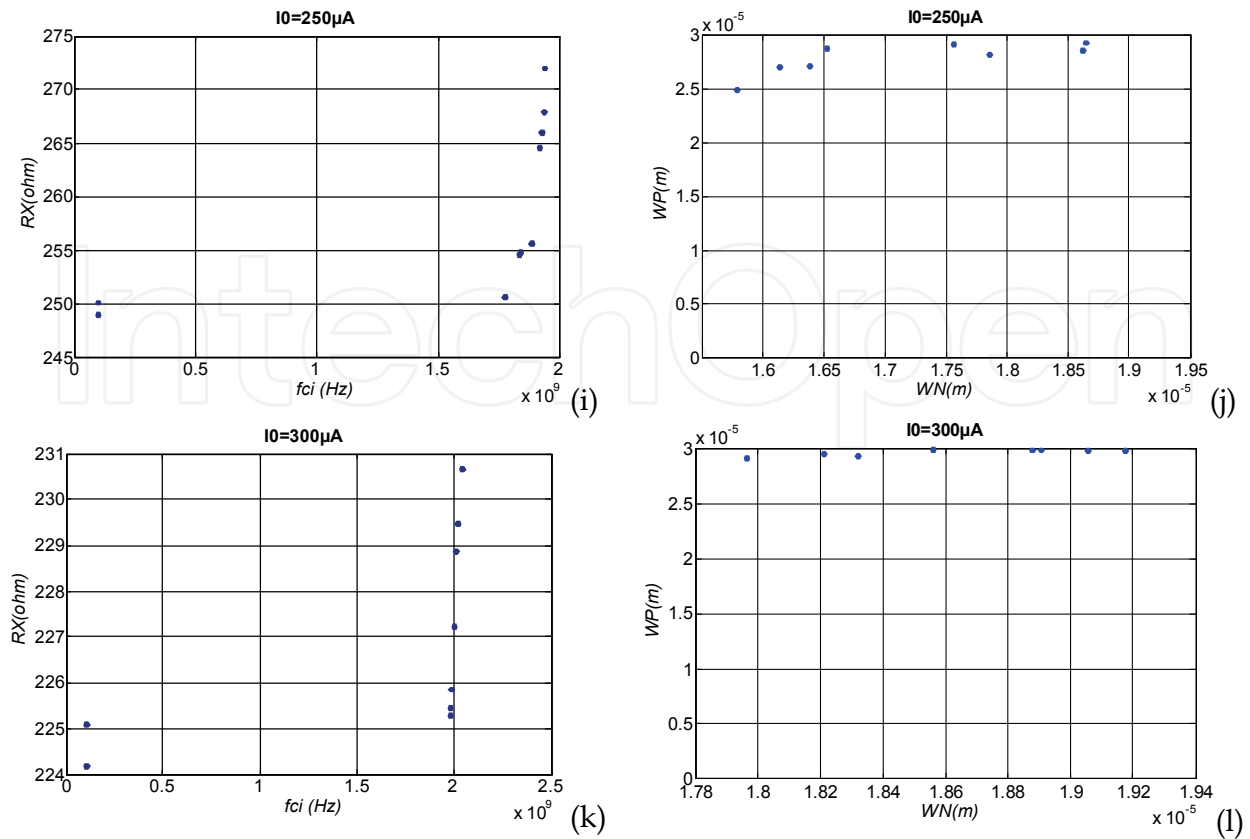


Figure 6. Pareto fronts and the corresponding variables for various bias currents

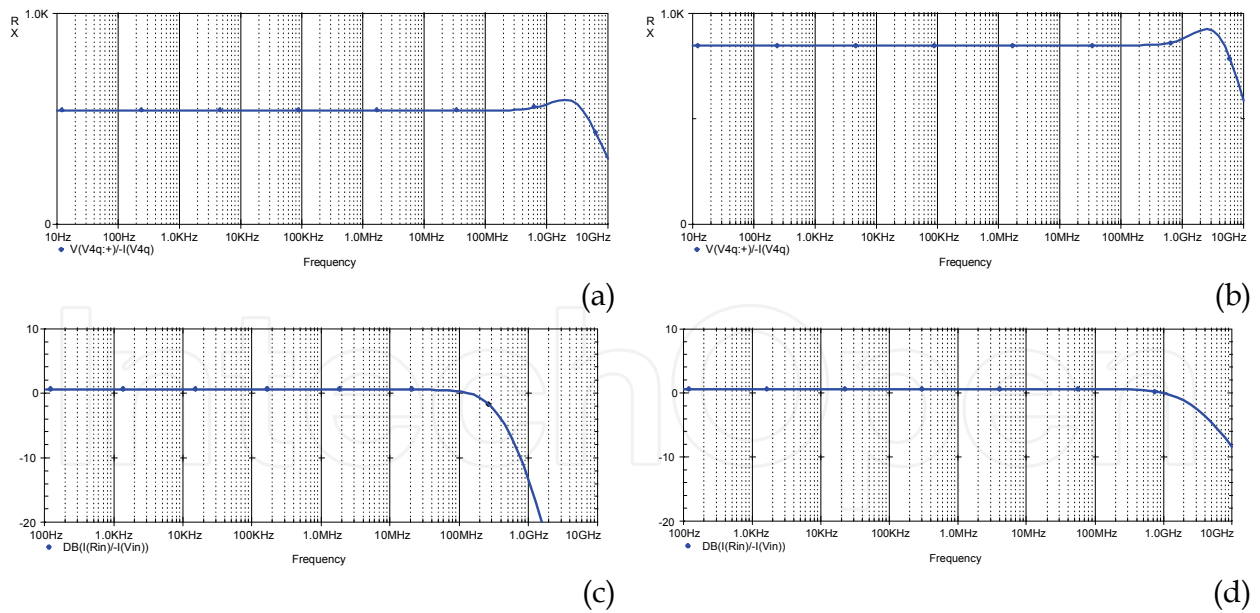


Figure 7. (R_X vs. frequency) Spice simulations

5. Conclusion

The practical applicability and suitability of the particle swarm optimization technique (PSO) to optimize performances of analog circuits were shown in this chapter. An

application example was presented. It deals with computing the Pareto trade-off surface in the solution space: parasitic input resistance vs. high current cut-off frequency of a positive second generation current conveyor (CCII+). Optimal parameters (transistors' widths and lengths, and bias current), obtained thanks to the PSO algorithm were used to simulate the CCII+. It was shown that no more than 1000 iterations were necessary for obtaining 'optimal' solutions. Besides, it was also proven that the algorithm doesn't require severe parameter tuning. Some Spice simulations were presented to show the good agreement between the computed (optimized) values and the simulation ones.

6. Appendix

In the analogue sizing process, the optimization problem usually deals with the minimization of several objectives simultaneously. This multi-objective optimization problem leads to trade-off situations where it is only possible to improve one performance at the cost of another. Hence, the resort to the concept of Pareto optimality is necessary.

A vector $\theta = [\theta_1 \dots \theta_n]^T$ is considered superior to a vector $\psi = [\psi_1 \dots \psi_n]^T$ if it dominates ψ , i.e., $\theta \prec \psi \Leftrightarrow \forall_{i \in \{1, \dots, n\}} (\theta_i \leq \psi_i) \wedge \exists_{i \in \{1, \dots, n\}} (\theta_i < \psi_i)$

Accordingly, a performance vector f^* is Pareto-optimal if and only if it is non-dominated within the feasible solution space \mathfrak{S} , i.e., $\neg \exists_{f \in \mathfrak{S}} f \prec f^*$.

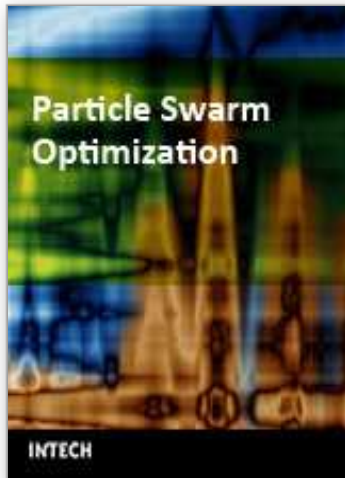
7. References

- Aarts, E. & Lenstra, K. (2003) Local search in combinatorial optimization. Princeton University Press.
- Banks, A.; Vincent, J. & Anyakoha, C. (2007) A review of particle swarm optimization. Part I: background and development, *Natural Computing Review*, Vol. 6, N. 4 December 2007. DOI 10.1007/s11047-007-9049-5. pp. 467-484.
- Basseur, M.; Talbi, E. G.; Nebro, A. & Alba, E. (2006) Metaheuristics for multiobjective combinatorial optimization problems : review and recent issues, report n°5978. National Institute of Research in Informatics and Control (INRIA). September 2006.
- Bellman, R. (2003) Dynamic Programming, Princeton University Press, Dover paperback edition.
- BenSalem, S; M. Fakhfakh, Masmoudi, D. S., Loulou, M., Loumeau, P. & Masmoudi, N. (2006) A High Performances CMOS CCII and High Frequency Applications, *Journal of Analog Integrated Circuits and Signal Processing*, Springer US, 2006. vol. 49, no. 1.
- Bishop, J.M. (1989) Stochastic searching networks, Proceedings of the IEE International Conference on Artificial Neural Networks. pp. 329-331, Oct 1989.
- Bonabeau, E.; Dorigo, M. & Theraulaz, G. (1999) Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press.
- Chan, F. T. S. & Tiwari, M. K. (2007) Swarm Intelligence: focus on ant and particle swarm optimization, *I-Tech Education and Publishing*. December 2007. ISBN 978-3-902613-09-7.
- Clerc, M. Particle swarm optimization (2006), *International Scientific and Technical Encyclopedia*, ISBN-10: 1905209045.

- Conn, A. R.; Coulman, P. K.; Haring, R. A.; Morrill, G. L.; Visweswariah, C. (1996), Optimization of custom MOS circuits by transistor sizing, *Proceedings of ICCAD96*, pp. 174-190, November, 1996. San Jose.
- Cooren, Y.; Fakhfakh, M. , Loulou, M. & Siarry, P. (2007) Optimizing second generation current conveyors using particle swarm optimization, *Proceedings of the 19th IEEE International Conference on Microelectronics*. December 29-31, 2007. Cairo, Egypt.
- Dastidar, T.R.; Chakrabarti, P. P. & Ray, P. (2005). Synthesis System for Analog Circuits Based on Evolutionary Search and Topological Reuse, *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 2, April 2005. pp. 211-224, ISSN 1089-778X.
- Doig, A.H. Land (1960), An automatic method for solving discrete programming problem, *Econometrica*, Vol. 28, pp. 497.
- Dorigo, M.; DiCaro G. & Gambardella, L. M. (1999) Ant algorithms for discrete optimization, *Artificial life journal* Vol. 5. pp. 137-172.
- Dréo, J.; Petrowski, A.; Siarry, P.; Taillard, E. (2006) *Metaheuristics for Hard Optimization: methods and case studies*, Springer. ISBN: 354023022X.
- Fakhfakh M., Loulou M. & Tlelo-Cuautle E. (2007) Synthesis of CCII and Design of Simulated CCII based Floating Inductances, *Proceedings of the 14th IEEE International Conference on Electronics, Circuits and Systems*. December 2007. Marrakech, Morocco.
- Ferri, G.; Guerrinin, N. & Piccirilli, M. C. (2002) Low voltage current conveyor-based biquad filter, *Proceedings of the IEEE International Symposium on Industrial Electronics*. Vol. 4. pp. 1331-1334.
- Glover, F. (1989) Tabu search- part I, *ORSA Journal on Computing*. Vol. 1, N° 3. Summer 1989.
- Glover, F. (1990) Tabu search- part II, *ORSA Journal on Computing*. Vol. 2, N° 1. Winter 1990.
- Gray, P. R.; Meyer, R. G.(1982) MOS operational amplifier design-a tutorial overview, *IEEE Solid-State Circuits*, Vol. 17, Issue 6, December 1982. pp. 969-982.
- Grimbleby, J. B. (2000) Automatic analogue circuit synthesis using genetic algorithms, *Proceedings of the IEE Circuits, Devices and Systems*, December 2000, Vol. 147, Issue: 6. pp: 319-323
- Kennedy, J & Eberhart, R. C. (1995) Particle swarm optimization, *Proceedings of the IEEE International Conference On Neural Networks*, pp 1942-1948, WA, Australia.
- Kirkpatrick, S.; Gelatt, C. D. & Vecchi, M. P. (1983) Optimization by simulated annealing, *Journal of Science*, 220:671-220-680.
- Medeiro, F.; Rodríguez-Macías, R.; Fernández, F.V.; Domínguez-astro, R.; Huertas, J.L.& Rodríguez-Vázquez, A. Global Design of Analog Cells Using Statistical Optimization Techniques, *Analog Integrated Circuits and Signal Processing*, Vol. 6, No. 3, November 1994. pp. 179-195, ISSN 0925-1030.
- Nelder, J.A. & Mead, R. (1965) A simplex method for function optimization, *Computer journal*, Vol. 7, pp. 308-313.
- Rajput, S. S. & Jamuar, S. S. (2007) Advanced applications of current conveyors: a tutorial, *Journal of active and passive electronic devices*, Vol 2, pp 143-164.
- Schmid, H. (2000) Approximating the Universal Active Element, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 47, N°. 11, Nov. 2000.
- Scniederjans, M.J.(1995) Goal Programming methodology and applications, Kluwer Publishers.

- Sedra, A. & Smith, K. C. (1970) A second generation current conveyor and its applications, *IEEE Transactions on Circuit Theory*, pp. 132-134, February 1970.
- Siarry(a), P.; Berthiau, G.; Durdin, F. & Haussy, J. (1997) Enhanced simulated annealing for globally minimizing functions of many-continuous Variables, *ACM Transactions on Mathematical Software*, Vol. 23, N°. June 1997. pp 209-228.
- Siarry(b), P.; Michalewicz, Z. (2007) *Advances in Metaheuristics for Hard Optimization*, Springer; December 2007. ISBN 3540729593.
- Smith, K. C. & Sedra, A. (1968) The current conveyor-a new circuit building block, *Proceedings of the IEEE (Letters)*, vol 56, pp. 1368-1369, August 1968.
- E. G. Talbi, E. G. (2002) A Taxonomy of Hybrid Metaheuristics, *Journal of Heuristics*, N°8 pp 541-564.
- Tlelo-Cuautle, E.; Duarte-Villaseñor, M.A.(2008), Evolutionary electronics: automatic synthesis of analog circuits by GAs, In *Success in Evolutionary Computation, Series: Studies in Computational Intelligence*, Yang, A.; Shan, Y. & Bui, L. T. (Eds.), pp. 165-187, Vol. 92, ISBN: 978-3-540-76285-0, Springer-Verlag, Berlin.
- Toumazou, C. & Lidgley, F. J. (1993), In *Analog IC Design: The current mode approach*, Haigh, D. G. (Ed.), *IEEE circuit and systems series 2*.
- Tulunay, G. & Balkir, S. (2004) A Compact Optimization Methodology for Single Ended LNA, *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 2004*.

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:

Mourad Fakhfakh, Yann Cooren, Mourad Loulou and Patrick Siarry (2009). A Particle Swarm Optimization Technique used for the Improvement of Analogue Circuit Performances, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from:
http://www.intechopen.com/books/particle_swarm_optimization/a_particle_swarm_optimization_technique_used_for_the_improvement_of_analogue_circuit_performances

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.

IntechOpen

IntechOpen