Int. J. Electron. Commun. (AEÜ) 66 (2012) 732–740

Contents lists available at SciVerse ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.de/aeue

Analog circuit sizing via swarm intelligence

R.A. Vural*, T. Yildirim

Yildiz Technical University, Department of Electronics and Communication Engineering, Davutpasa Campus, 34220 Esenler, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 8 September 2011
Accepted 5 January 2012

Keywords:
CMOS analog integrated circuit sizing
Particle swarm optimization
Circuit design automation
Area oriented optimization

ABSTRACT

Together with the increase in electronic circuit complexity, the design and optimization processes have to be automated with high accuracy. Predicting and improving the design quality in terms of performance, robustness and cost is the central concern of electronic design automation. Generally, optimization is a very difficult and time consuming task including many conflicting criteria and a wide range of design parameters. Particle swarm optimization (PSO) was introduced as an efficient method for exploring the search space and handling constrained optimization problems. In this work, PSO has been utilized for accommodating required functionalities and performance specifications considering optimal sizing of analog integrated circuits with high optimization ability in short computational time. PSO based design results are verified with SPICE simulations and compared to previous studies.

1. Introduction

Analog integrated circuit (IC) design is a challenging process which involves the characterization of complex tradeoffs between nonlinear objectives and also satisfying required constraints. Those objectives are comprised of design parameters which are ideally accepted as variables and optimum solution set is searched. However, as the circuit complexity increases the search space expands such that obtaining the optimal combination of design parameters by hand becomes a time consuming and unaffordable process. Considering CMOS IC design process, there are several relations that should hold between length, width and width/length ratios of MOS transistors to ensure that the search space is smooth and the optimization process is reliable. Therefore, efficient optimization methods are required for automation of optimal sizing of CMOS analog IC design.

Classical optimization approaches are either deterministic or statistical-based techniques. Deterministic methods, such as Simplex [1], Branch and Bound [2], Goal Programming [3], and Dynamic Programming [4] are effective only for small size problems. These optimization techniques impose several limitations for multi-criteria constrained problems due to their inherent solution mechanisms and their tight dependence on the algorithm parameters. Most of the optimization problems require different types of variables, objective and constraint functions simultaneously in their formulation. Statistical methods generally start with finding a "good" DC quiescent point, which is provided by the skilled analogue designer. Following, 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 [5]. Therefore, classic optimization procedures are generally not adequate for optimal sizing of analog integrated circuits.

Heuristics are necessary to solve big size problems and/or with many criteria [6]. They can be adapted to suit specific problem requirements. Even though they do not guarantee to find in an exact way the optimal solution, they provide good approximation of it within an acceptable computing time [7]. Some mathematical heuristics that were previously utilized were Local Search [8], Simulated Annealing (SA) [9,10], Tabu Search (TS) [11,12], Genetic Algorithms (GA) [13,14], etc.

However, efficiency of these techniques is highly dependent on the algorithm parameters, the dimension 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 generally require long computation time when complexity of the problem increases. 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 behavior of decentralized, self-organized systems. It is known as Swarm Intelligence (SI) [15]. SI systems typically employ a population of simple agents 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 behavior. Most famous such SI techniques are Ant Colony Optimization (ACO) [16],

* Corresponding author.
E-mail addresses: racar@yildiz.edu.tr (R.A. Vural), tulay@yildiz.edu.tr (T. Yildirim).

1434-8411/$ – see front matter © 2012 Elsevier GmbH. All rights reserved.
doi:10.1016/j.aeue.2012.01.003

Artificial Bee Colony (ABC) optimization [17] and Particle Swarm Optimization (PSO) [18,19]. PSO has been in existence for almost a decade, which is a relatively short period when compared to some of the well known evolutionary computation paradigms and has been shown to offer good performance in various application domains [20].

Above mentioned optimization methods are incorporated into analog computer-aided design (CAD) tools for optimal sizing of complex ICs together with topology selection [21] and actual circuit layout [22]. Historically, researchers developed two mainstreams of analog automation methodologies. One of the early approaches uses optimization-based method which optimizes a set of performance constraints characterized by complicated tradeoffs and makes repetitive use of detailed circuit simulator embedded in the inner loop of optimization engine. These techniques require many iterations to adjust transistor sizes and optimization engine needs to evaluate corresponding performance at each cycle. Second approach is equation-based method which is based on inverse process of circuit analysis technique. Since sizing of a circuit is done mathematically, the automation is much faster while accuracy is not as good as the first approach due to the simplified device equations and approximations [23,24]. Comparison of previously proposed analog CAD tools is given in Table 1 [22–31]. Among the CAD tools tabulated in Table 1, DELIGHT.SPICE, STAIC and OPASYN utilize classical optimization techniques while IDAC, MAELSTROM, ASTRX/OBLX, ASLIC, and OASYS are heuristic based systems. Kruiskamp and Leenaerts [32] developed a GA based CMOS operational amplifier synthesizer (DARWIN) for topology selection and circuit sizing. In [33], sizing rules method is proposed for CMOS and bipolar analog IC synthesis. Sripramong and Toumazou [34] introduced an automated circuit design system for the evolution and subsequent invention of CMOS amplifiers. This system utilized genetic programming for evolving new circuit topologies and current-flow analysis for screening and correcting circuits. In [35], it was proved that CMOS op-amp design can be approximated as convex optimization problem that can be solved using geometric programming techniques. In [36], an evolution-based methodology named memetic single-objective evolutionary algorithm is developed for automated sizing of high-performance analog IC circuits. Guerra-Gomez et al. [37] proposed multi-objective evolutionary algorithm based on decomposition (MOEA/D) for optimization of second generation current conveyors (CCII). Mentioned system uses HSPICE as circuit evaluator. Considering optimal CCII design without any circuit evaluator; a multi-objective heuristic [38,39] and PSO algorithm [40–42] are utilized by formulating the requirements for the design of CCII in terms of boundaries on performance functions. Tawdross and König [43] investigated PSO as an alternative to GA for field programmable analog scalable device array reconfiguration. For this purpose an operational amplifier with particular design constraints was designed using PSO taking into different external influences such as high temperature and fabrication faults. Having successful results authors extended their PSO based dynamic hardware design environment to functional block level [44]. A 3-bit ADC structure is developed using previously designed op-amps and resistors. In [45], PSO algorithm is extended to a hierarchical scheme for automatic sizing of low power analog circuits where simulation of circuits is performed with Cadence Spectre. Tulunay and Balkir [46] proposed an automatic synthesis tool of a cascade low noise amplifier (LNA). In [42], PSO technique is utilized for optimal sizing of CMOS LNA with inductive degeneration design. Choi and Allstot [47] developed a SA based synthesis tool that includes an adaptive tunneling mechanism and post-optimization sensitivity analysis with respect to design, process and temperature variations. A detailed investigation about the state of the art in applying EAs for the synthesis and sizing of analog ICs is presented in [48].

This main objective of this study is to explore Particle Swarm Optimization algorithm on analog circuit design automation. PSO-based method is applied to two analog integrated circuit design problems with particular technology parameters. The problem considered in this work is the optimal CMOS transistor sizing for minimum area oriented optimization, which is only a part of a complete analog circuit CAD tool. Other parts which are beyond the scope of this work are the topology selection and actual circuit layout. The optimal transistor sizing of the CAD process remains between these two tasks. As reported in the literature, simulation-based optimization technique requires very long execution time and equation-based methods are less accurate than the former method. Therefore, optimization methods with high accuracy and short computation time are necessary for analog circuit design automation. PSO as a global optimization method has fewer primitive mathematical operators than in GA (e.g. reproduction, mutation and crossover) and those mathematical operations require more fine-tuning of own parameters which leads to longer computation time as explained in Section 2. Section 3 describes analog integrated circuit structures and states design specifications used in optimal sizing task. PSO-based method for integrated circuit design is investigated in Section 4. Following, Section 5 provides simulation results of the proposed method, comparable to previous methods, which are validated with SPICE simulator. Finally, Section 6 concludes with a discussion of PSO based design results and suggests possible extensions.

2. Particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation method based on the social behavior, movement and intelligence of swarms searching for an optimal location in a multidimensional search area [18]. The approach uses the concept of population and a measure of performance similar to the fitness value used with evolutionary algorithms. Population consists of potential solutions called particles. Each particle is initialized with a random position value. In each iteration of simulation, the fitness function is evaluated by taking the current position of the particle in the solution space and two best values (p_{best}, g_{best}). Personal best value, p_{best}, is the location of the best fitness value obtained

Table 1
The error rate and synthesis time of various analog CAD tools [23,24].

Tool	Synthesis method	Error	Synthesis time
IDAC [25]	Equation-based	15%	Few seconds
OASYS [26]	Equation-based	25%	Few seconds
ISAID [27]	Equation-based + post optimization	14%	Not reported
STAIC [22]	Equation-based	24%	3 min
DELIGHT.SPICE [28]	Optimization-based (circuit simulator)	0%	18 h
MEALSTROM [29]	Optimization-based (circuit simulator)	0%	3.6 h
ASTRX/OBLX [30]	Optimization-based (AWE + equations)	30%	11.8 h
OPASYN [31]	Optimization-based (equations)	20%	1 min
ASLIC [24]	Equation-based	15–20%	Few seconds