



# A hierarchy in mutation of genetic algorithm and its application to multi-objective analog/RF circuit optimization

Satyabrata Dash<sup>1</sup> · Deepak Joshi<sup>1</sup> · Ayushparth Sharma<sup>2</sup> · Gaurav Trivedi<sup>1</sup>

Received: 4 August 2017 / Revised: 28 October 2017 / Accepted: 29 November 2017 / Published online: 6 December 2017  
© Springer Science+Business Media, LLC, part of Springer Nature 2017

## Abstract

This paper presents a multi-objective analog circuit design optimization tool using genetic algorithm based on hierarchical mutation scheme. The idea is to improve the convergence and diversity of genetic algorithm by incorporating hierarchy during polynomial mutation operation. In this regard, a theoretical framework of proposed genetic algorithm is presented using Markov chain principle. To investigate the effectiveness of hierarchy in polynomial mutation operator, the scheme is compared with six different mutation strategies. Experiments are performed for different function evaluations to evaluate the performance of hierarchical polynomial mutation operator. Further, to showcase the improvement in genetic algorithm, numerous experiments are performed on twelve different test functions and two design examples. The proposed genetic algorithm shows competitive performance over other standard optimization techniques in terms of both convergence and diversity of solutions.

**Keywords** Circuit optimization · Genetic algorithm · Multi-objective optimization · Polynomial mutation

## 1 Introduction

Analog circuit synthesis process is typically characterized by selecting a set of design parameters and topology. With the rapid progress in VLSI technology, manual design of an analog circuit is often confined to the usage of design automation for a fixed topology in finding a solution space, satisfying a set of design constraints. Due to high degree of nonlinearity and interdependence among design parameters, extensive computation is required to evaluate various design parameters during circuit design process. Although it is beneficial to reduce the computation cost associated

with generating design performance estimates by modeling the circuit design problem in the form of a linear programming problem and solving it using efficient optimization techniques, relevant emphasis is needed to analyze tradeoffs among various design specifications.

Many optimization techniques have been presented in literature to address the problem associated with analog circuit synthesis. One approach is to analyze analog circuit design using geometric programming [1], which derives a set of specific equations by considering design constraints before applying it to the design process. The look-up table approach [2] is used to approximate circuit equations and replace device circuit models for accurate circuit simulation during optimization. With increase in number of design parameters and device dimension, the size of the table grows exponentially and it becomes difficult to generate and store such large tables efficiently. Further, there are methods, such as goal attainment method, which makes use of weights to control the tradeoffs among design specifications [3]. Further, nature inspired optimization techniques, such as genetic algorithm [4, 5], particle swarm optimization [6], differential evolution [7] etc. solve cost functions involving design specifications, which is translated from modeling the behavior of analog circuit design

---

✉ Satyabrata Dash  
satyabrata@iitg.ernet.in

Deepak Joshi  
d.joshi@iitg.ernet.in

Ayushparth Sharma  
parth.70121@gmail.com

Gaurav Trivedi  
trivedi@iitg.ernet.in

<sup>1</sup> Indian Institute of Technology Guwahati, Guwahati, India

<sup>2</sup> LNM Institute of Information Technology, Jaipur, India

subject to specific set of design constraints. Such nature inspired algorithms function by emulating certain natural phenomenon to obtain global optimum solution and demand the design of circuit equations to be more realistic by considering suitable design parameters.

Genetic algorithm (GA) has become one of the popular optimization techniques in recent years after introduced by Holland [8]. Mostly, GA has been employed as a stochastic procedure to obtain global optimum solution for different combinatorial optimization problems such as scheduling problems [9], traveling salesman problem [10, 11] and in machine learning [12]. Further, GA has been applied to solve both single objective, multi-objective and many objective problems in literature [13, 14]. Nondominated sorting genetic algorithm (both NSGA-II and NSGA-III) has been presented to handle multi- and many-objective problems including the ability to find best possible solution with acceptable convergence. Although other approaches satisfy themselves by compensating for suboptimal performance in certain cases, NSGA-II has an appealing feature of overcoming difficulties in traditional multi-objective circuit optimization.

The application of NSGA-II has not been seriously addressed in multi-objective circuit optimization as per our knowledge. Therefore, in this paper a hierarchical version of NSGA-II (*hNSGA-II*) is presented to perform analog circuit optimization using hierarchy in generation of new population during polynomial mutation operation. At first, the multidimensional search space is updated with new parent population using hierarchical polynomial mutation strategy. Later nondominated sorting is used to obtain local optimal solutions among neighborhood in a single hierarchical mutation growth. After this phase, the local individuals are checked for distinct fitness values for diversity preservation. Once the search for unique fitness values is completed, the local best individuals are merged with parent population similar to NSGA-II framework. Further, the proposed *hNSGA-II* framework facilitates the optimization process to achieve required Pareto front [15] by carrying out repeated calculations, and evaluates circuit design equations comfortably. This paper also develops a theoretical framework for *hNSGA-II* based on the asymptotic states of a homogeneous Markov chain model and proves the existence of stationary distribution with constant nonzero hierarchical mutation probability value (strictly positive). Further, a strong ergodicity bound is established to ensure the convergence to limit distribution asymptotically. For showcasing the effectiveness of proposed hierarchical scheme in polynomial mutation operation, six different mutation operators are incorporated in *hNSGA-II* framework and compared for convergence and diversity of Pareto optimal solutions. Further, to evaluate the performance of proposed *hNSGA-II* framework, several

experiments are performed on twelve multi-objective test functions and two design examples. The effectiveness of *hNSGA-II* is shown by comparing it with standard optimization algorithms.

The rest of the paper is organized as follows. Overview of multi-objective optimization problem and NSGA-II tool are presented in Sect. 2. The proposed *hNSGA-II* framework is showcased in Sect. 3. A detailed performance analysis of proposed framework is shown by running extensive simulations on standard benchmarks and by comparing the results with other conventional optimization techniques in Sect. 4. The effectiveness of proposed framework is shown by optimizing design specifications for two design examples in Sect. 5. Section 6 concludes the manuscript.

## 2 Background

### 2.1 Multi-objective optimization

Multi-objective optimization (MOP) is a method centered on various conditions to satisfy behavior of contradictory objectives or cost functions simultaneously. Contrary to single objective optimization, the solution obtained by MOP is not confined to a single point solution, but a set of balanced solutions known as Pareto optimal solutions, which when plotted forms a Pareto front (PF) in the multi-objective search space [15]. The novelty of MOP using any optimization technique is to evaluate a set of solutions close to PF without loss of convergence and diversity properties. These properties ensure to deal with non-dominated solutions [13], while carrying out continuous design space exploration. Given an  $n$ -dimensional Euclidean space  $\mathbf{R}^n$  and objective function vector  $\mathbf{F}(\mathbf{x})$ , the multi-objective optimization attempts to find the solution  $\mathbf{F}(\mathbf{x}')$  satisfying,

$$\begin{aligned} \mathbf{F}(\mathbf{x}') &= \min_{\mathbf{x} \in X} \mathbf{F}(\mathbf{x}) \\ &= \min_{\mathbf{x} \in X} (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \end{aligned} \quad (1)$$

where  $\mathbf{F} : \mathbf{R}^n \rightarrow \mathbf{Q}^m$  consists of  $m$  real-valued objective functions and  $\mathbf{Q}^m$  is the objective space [15]. Since the objective functions in (1) often contradict with each other, it is impossible to find a single solution in  $\mathbf{R}^n$  that minimizes all objectives simultaneously. Therefore, multi-objective optimization finds the best tradeoffs among objective functions in terms of Pareto optimality, which is guided by a dominance relation between solutions. Let  $a, b \in \mathbf{Q}^m$ ,  $a$  dominates  $b$  if and only if  $a_i \leq b_i \forall i \in \{1, \dots, m\}$  and  $a_k < b_k$  for at least one index  $k \in \{1, \dots, m\}$  [15]. A point  $x' \in \mathbf{Q}^m$  is said to be Pareto optimal [15] if there is no point in  $\mathbf{R}^n$  such that  $\mathbf{F}(\mathbf{x})$  dominates  $\mathbf{F}(\mathbf{x}')$ , where  $\mathbf{F}(\mathbf{x})$  is a vector of solutions called Pareto optimal

vector if it contains only Pareto optimal solutions. The set of all Pareto optimal objective vectors forms PF.

## 2.2 Nondominated sorting genetic algorithm (NSGA-II)

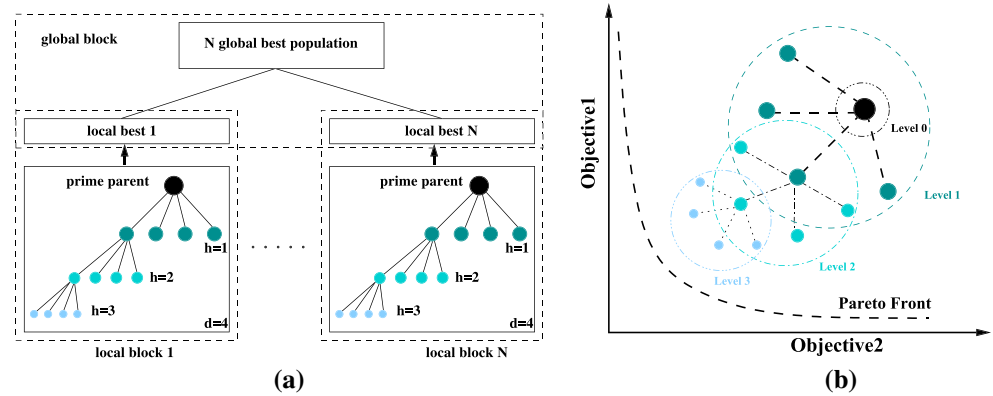
NSGA-II employs nondominated sorting approach for fitness assignment among individual solutions to solve multi-objective optimization problems (MOPs) [15]. The algorithm starts with a random generation of initial population (range of design alternatives for design variables) of  $N$  numbers. The current parent population ( $P_t$ ) at  $t$ th generation is used to produce offspring population ( $Q_t$ ) with population size  $N$  by using polynomial mutation and simulated binary crossover operator [13]. The two populations  $P_t$  and  $Q_t$  are merged together to form a mixed population  $R_t$  (of population size  $2N$ ). The best  $N$  numbers are selected from  $R_t$  pool using nondominated sorting approach based on dominance principle. Nondominated solutions are selected from  $R_t$  to fill for a population size of  $N$  in next generation. The remaining solutions in next generation pool are selected by employing a crowding distance scheme [13], i.e., the solutions having largest crowding distance values are chosen for next generation to maintain a desired diversity.

## 3 Hierarchical mutation based genetic algorithm

A framework for hierarchical version of NSGA-II ( $h$ NSGA-II) is presented in this section. The proposed framework is based on the logic of conventional NSGA-II. Original NSGA-II makes use of a fast nondominated sorting approach with computational complexity  $O(kN^2)$  [13], where  $k$  is the number of objectives and  $N$  is the population size. Along with this elitist-preserving approach, NSGA-II uses crowding distance scheme to estimate the density of solutions for any specific PF. However, the formulation of this scheme becomes unstable when two or more solutions share a common fitness. Although NSGA-II keeps track of convergence by giving emphasis on nondominated solutions, there is scope of improvement in this category by employing suitable mutation or crossover strategies. The objective of our proposed approach is to improve the convergence property by incorporating hierarchy in the existing polynomial mutation scheme of NSGA-II. A generic hierarchical approach is presented for generation of child population during mutation operation, i.e., each child individual is formed in local neighborhood of its siblings along with parent population. The hierarchical approach for mutation operation

is illustrated in Fig. 1(a). The child population is generated from corresponding parent population in a hierarchical structure, which forms local blocks of manageable size. Let us consider the generation of  $N$  local blocks and subsequent formation of one global block combining  $N$  local blocks and previous population. Each of the  $N$  local blocks is modeled as a tree like structure (having height  $h$ ) with child population forming different branches. The first prime individual in a local block is called a *prime parent*. Prime parent generates a set of child individuals in first level ( $h = 1$ ) and the individuals in first level generate child individuals for second level ( $h = 2$ ) through mutation operation. Hence, the individuals in first level act as parents for individuals of second level (sub-parents) and so on. In such manner, child population in each local block is generated using polynomial mutation operator and the process continues till maximum height of the tree ( $h_{\max}$ ) is achieved. The number of child individuals of prime parent and each sub-parent is decided by the degree of branching  $d$ , and a total of  $\beta$  individuals are generated in each local block including the prime parent during each generation. The generated local best solutions in  $n$ th local block having  $d = 4$  moving towards PF in each level, is shown in Fig. 1(b), where  $n \in N$ . Further, all child individuals in a local block along with prime parent are sorted and ranked using nondominated sorting approach. The individuals having highest ranks in each local block are selected as local best individuals. The local best individuals are pulled to global block to generate the offspring population. The offspring population is merged with parent population to generate a population having  $2N$  individuals. These individuals are again sorted and ranked using nondominated sorting approach to select the best individuals for next generation. However, to preserve the diversity during selection, it is necessary to select distinct local best individuals before being pulled to global block without affecting the convergence property. Instead of performing the ranking through traditional crowding distance approach, we propose to perform an indirect exclusive selection by performing hierarchical mutation operation iteratively. For an exclusive selection, at first  $N$  local best individuals are sorted in descending order of crowding distance. All local best individuals are checked for unique fitness and the individuals having unique fitness values ( $\epsilon$ ) are retained in the local best pool. The remaining  $N - \epsilon$  local best individuals are discarded and hierarchical mutation operation is carried out in corresponding  $N - \epsilon$  local blocks to generate new local best individuals. The process is continued till  $N$  local best individuals with distinct fitness values are generated. As the population in each local block is more than one ( $\beta > 1$ ), where  $\beta \leq \sum_i^h d^i$ , the probability of generating local best individuals having

**Fig. 1** **a** General model of hierarchical mutation strategy, **b** hierarchical growth for  $h = 3$  and  $d = 4$  showing growth towards true PF



distinct fitnesses is more as compare to traditional crowding distance scheme. In this regard, it accounts for the instability of crowding distance scheme in choosing common fitness values. While performing an indirect exclusive selection using hierarchical mutation operation, the number

of individuals in each local block also play an important role in convergence of solutions. A detailed theoretical framework of  $h$ NSGA-II along with convergence property is described in following subsections.

---

**Algorithm 1** HIERARCHICALMUTATION ( $P_t$ ) procedure

---

```

1: Initialize height ( $h$ ) of each local block
2: Initialize RAND_MAX ( $R_{max}$ ) for random number generation
3:  $d \leftarrow \lceil \text{RAND}(R_{max}) \rceil$   $\triangleright R_{max} \in [1, 4]$ 
4:  $size \leftarrow N$ 
5: for  $i = 1$  to  $size$  do
6:    $j \leftarrow 1$ 
7:   while  $j \leq h$  do
8:     for  $l = 1$  to  $d^j$  do
9:        $C_t \leftarrow \text{POLYNOMIALMUTATION}(P_t)$   $\triangleright C_t \rightarrow$  local block population
10:    end for
11:     $j \leftarrow j + 1$ 
12:  end while
13:   $H_t \leftarrow \text{NONDOMINATEDSORT}(C_t)$ 
14:   $cbest_t^i \leftarrow \text{ELITECHILD}(H_t)$ 
15: end for
16:  $j \leftarrow 0$ 
17: for  $i = 1$  to  $N$  do
18:   for  $k = 1$  to  $N$  do
19:    if  $|cbest_t^i| \neq |cbest_t^k|$  then
20:      if  $|cbest_t^k| \in Q_t$  then
21:        continue
22:      else
23:         $Q_t \leftarrow Q_t \cup cbest_t^k$ 
24:      end if
25:    else
26:       $j \leftarrow 1$ 
27:      while  $j \leq h$  do
28:        for  $l = 1$  to  $d^j$  do
29:           $C_t \leftarrow \text{POLYNOMIALMUTATION}(P_t)$ 
30:        end for
31:         $j \leftarrow j + 1$ 
32:      end while
33:       $H_t \leftarrow \text{NONDOMINATEDSORT}(C_t)$ 
34:       $cbest_t^i \leftarrow \text{ELITECHILD}(H_t)$ 
35:      go to step 19
36:       $\varepsilon \leftarrow \varepsilon + 1$ 
37:    end if
38:  end for
39:   $\varepsilon \leftarrow 0$ 
40: end for

```

---

---

**Algorithm 2** *h*NSGA-II procedure

---

```

1: Generate initial parent population  $P_1$  of size  $N$ 
2:  $Q_1 \leftarrow \text{HIERARCHICALMUTATION}(P_1)$   $\triangleright Q_t \rightarrow$  offspring population
3: for  $t = 1$  to  $NGEN$  do
4:    $R_t \leftarrow Q_t \cup P_t$ 
5:    $F \leftarrow \text{NONDOMINATEDSORT}(R_t)$ 
6:    $P_{t+1} \leftarrow \phi$ 
7:    $j \leftarrow 1$ 
8:   while  $|P_{t+1}| + |F_j| \leq N$  do
9:      $\text{ASSIGNCROWDINGDISTANCE}(F_j)$ 
10:     $P_{t+1} \leftarrow P_{t+1} \cup F_j$ 
11:     $j \leftarrow j + 1$ 
12:     $\text{QUICKSORT}(F_j)$ 
13:   end while
14:    $Q_{t+1} \leftarrow \text{HIERARCHICALMUTATION}(P_{t+1})$ 
15: end for

```

---

**3.1 Computational complexity**

As *h*NSGA-II is based on the framework of conventional NSGA-II, the algorithm starts by initializing parent population  $P_1$  of first generation. The hierarchical mutation procedure described in Algorithm 1, starts with initializing local block of height  $h$  and random generation of degree  $d$  for generating a temporary offspring population  $C_t$ . Once the number of offsprings in each level  $d$  is defined, the algorithm follows polynomial mutation operator to generate offspring population in each local block. This requires  $O(hd^j)$ , where  $j \in [1, h]$  number of computations in each block and a total of  $O(hd^h(MN))$  number of computations (as  $j$  can go upto  $h$ ) for generating the local offspring population  $C_t$  with  $M$ -dimensional objectives. The non-dominated sorting in line 13 of Algorithm 1 having  $N$  population size requires  $O(MN^2)$  computations [13]. As the search for unique fitness value among elite local individuals is recursive, it requires  $O(\varepsilon(MN^2))$  computations with  $M$ -dimensional objectives. Once the local elite individuals are generated with unique fitness values, they are pulled to the global block, which forms the offspring population ( $Q_t$ ) of  $t$ th generation. This  $Q_t$  offspring population is combined with  $P_t$  to generate  $R_t$  having population size of  $2N$ . The nondominated sorting described in line 5 of Algorithm 2 having  $2N$  population size requires  $O(M(2N)^2)$  computations [13]. Crowding distance assignment in line 9 requires  $O(k(2N)\log(2N))$  computations in the worst-case and quicksort involves  $O(2N\log(2N))$  computations, which speaks for an overall complexity of  $O(M(\beta N + \varepsilon N^2))$  for *h*NSGA-II procedure, where  $1 \leq \beta \leq (\sum_i^h d^i)$ . With increase in the height of each block, the runtime of *h*NSGA-II increases with an advantage of reaching convergence at minimum generations. One of the reasons for this functional advantage is the selection of an elite offspring from a pool of randomly generated individuals spanning alternative search spaces. Further, this

ensures the evolution process to be promising as hierarchy in mutation operation facilitates the introduction of new offspring individual from random locations within the feasible space to preserve diversity in each generation.

**3.2 Theoretical framework**

This section presents a theoretical framework for *h*NSGA-II based on asymptotic states of a homogeneous Markov chain algorithm. As *h*NSGA-II is based on NSGA-II, it uses the same basic operators such as selection, crossover and mutation to create next generation from current population. The generation of successive populations can be viewed as a stochastic process with finite state space, and each operation carried out by the operators can be modeled as corresponding stochastic matrices. Further, the probabilistic changes of real parameters within a population using genetic operators can be formulated in the form of transition probability matrix  $P_r$  [16], which can be represented as the product of stochastic matrices,

$$P_r = P_s \cdot P_m \cdot P_c, \tag{2}$$

where  $P_s$ ,  $P_m$  and  $P_c$  denote intermediate transitions caused by selection, mutation and crossover operation, respectively. Let us consider each genetic operation to be performed in succession. It is assumed that proportional selection is performed with probability  $P_s$  on an initial population vector  $\bar{x} \in S$  having set of all possible population states  $S$ . The conditional probability  $P_s(\bar{y}|\bar{x})$  of selecting a solution vector  $\bar{y} \in S$  from initial population state vector  $\bar{x} \in S$  can be represented as [16],

$$P_s(\bar{y}|\bar{x}) = \frac{\prod_{k \in S} f(y_k)}{(\sum_{k \in S} f(y_k))^n}, \text{ if } \bar{y} \subset \bar{x}, \quad (\bar{y}, \bar{x}) \in S \tag{3}$$

$$= 0 \text{ otherwise}$$

Mutation operation is performed to maintain the genetic diversity in successive generations by making changes to the states of decision parameters (variables) of current

generation. As mutation operation is carried out independently on each individual decision variable by setting a user-defined mutation probability  $p_m$  to lower values (i.e.  $p_m \in \{0, 1\}$ ), the conditional probability  $P_m(\bar{y}|\bar{x})$  that the new population  $\bar{y} \in S$  resembles the initial population  $\bar{x} \in S$  after altering real values of variables at different decimal positions (mutation operation) can be aggregated to [16],

$$P_m(\bar{y}|\bar{x}) = p_m^{\text{poly}(x_k, y_k)} (1 - p_m)^{\text{poly}(x_k, y_k)}, \tag{4}$$

where  $\text{poly}(x_k, y_k)$  is the polynomial mutation operator<sup>1</sup> [17] to generate child individual  $y_k$  from parent  $x_k$ . Typically, crossover operation is performed to generate child solution from two or more parent solutions. Here, simulated binary crossover (*SBX*) operator is used for crossover operation in *hNSGA-II* because of the classic property to keep difference between child individuals proportionate to their corresponding parent individuals. *SBX* coefficient provides the flexibility to control generation of child population in search space, i.e. child individuals having high *SBX* coefficients are more probable to provide solutions close to optimum. Although *SBX* operator is used with crossover probability  $P_c \in [0, 1]$ , convergence analysis can be performed even without the knowledge of elements of transition crossover matrix as the choice of any specific operator does not affect the analysis.

The formal description of our proposed approach relies heavily on the hierarchical scheme of genetic evolution. The behavior of this scheme can be interpreted as a tree like homogeneous time bivariate Markov chain  $\{(K_t, G_t), t \geq 0\}$ , where  $K_t$  represents all individuals in the local block having height  $h = 2$  and degree  $d = 4$ , and  $G_t \in \mathbb{N}, G_t > 0$  as shown in Fig. 2. Further, the prime parent is denoted by  $\alpha$  and all other individuals in the local block are denoted by strings of integers in  $[1, \beta]$ , where  $\beta \leq \sum_i^h d^i$ . Moreover, the behavior of the hierarchical mutation scheme implies few restrictions on the Markov chain for proper functionality. All transitions are considered to be of zero probability with the following two exceptions.

1. At each step the process can make one transition to itself ( $K_{t+1} = K_t$ ) (no change in parent population after mutation operation) or,
2. Transition to one of its child ( $K_{t+1} = K_{t+s}$ , for some  $1 \leq s \leq d$ ) (change in parent population after mutation operation).

Local block of the *hNSGA-II* framework represents the evolution of child population from parent population using mutation operation. The transition probabilities of a parent to its children and to itself in one level, may be different

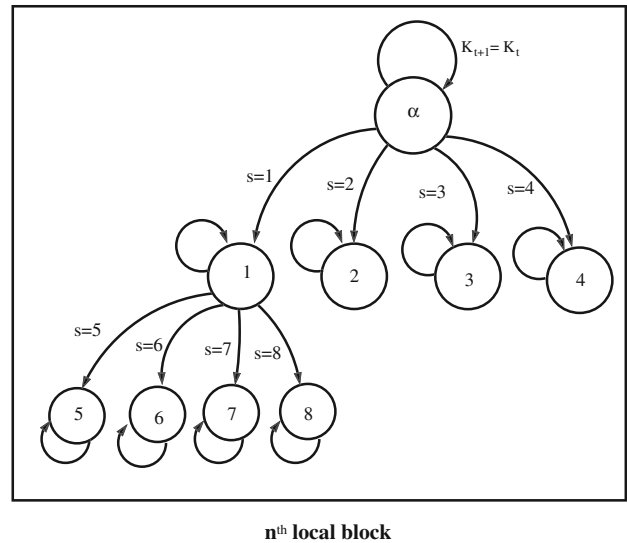


Fig. 2 Markov chain model of hierarchical strategy

even though both transitions are from a single individual. In view of this, Markov chain in each local block can be illustrated as the sum of probabilities of all possible sample paths in forward direction starting from prime individual and ending at child individuals of final level. Therefore, the transition probability matrix of mutation can be rewritten as a combination of intermediate transitions described in (4) of each probable states. The transition probability matrix of mutation at level  $h$  having degree  $d$  can be represented as,

$$P_m^{(h)}(\bar{y}|\bar{x}) = \sum_{l \in h} \left( \sum_{j \in d^l} p_m^{\text{poly}_j^l(x_k, y_k)} (1 - p_m)^{\text{poly}_j^l(x_k, y_k)} \right) \tag{5}$$

It can be pointed out that the hierarchical mutation operator holds the key for robust convergence of *hNSGA-II* because it exhibits the decision whether the algorithm achieves limiting distribution (strong ergodicity) [18] or not. With the increase in the height  $h$  of each local block, the number of randomly generated individuals increase allowing the hierarchical mutation operator to explore additional search points in the feasible region. This proposal of augmented upper bound in mutation operation facilitates sufficient conditions for strong ergodicity.

### 3.3 Convergence analysis

Considering the search properties of *hNSGA-II*, various conclusions can be made relating to the sufficient conditions for *hNSGA-II* converging to a stationary distribution.

**Proposition 1** Any state probability distribution  $\xi \in X$  of a homogeneous Markov chain is a stationary distribution in  $X$  if it satisfies the equality,  $\xi^T = \xi^T P_r$ , where  $P_r$  is the

<sup>1</sup> Polynomial mutation operation is considered as it is carried out using hierarchical scheme.

transition probability matrix and  $X$  is the state space containing all possible Markov states [16].

**Theorem 1** For  $p_m > 0$  and a strictly positive defined crossover and selection probability  $P_c, P_s$  respectively,  $h$  NSGA-II converges in limit towards a stationary distribution  $\xi$ , where  $\xi$  is strictly positive, i.e.  $\xi > 0$ , i.e.,

$$\lim_{r \rightarrow \infty} P_r = \begin{bmatrix} \xi \\ \xi \\ \vdots \\ \xi \end{bmatrix} \tag{6}$$

**Proof** Let  $(x_t, x_{t+1})$  be the given arbitrary states at any time point  $(t, t + 1)$  respectively before hierarchical mutation. The transition from  $x_t$  to  $x_{t+1}$  in one time step with appropriate probability  $P_m$  can be given by (5). Let us assume the corresponding relative fitness be  $f_{(t,t+1)}$ . Depending upon the number of states in each block, there are two possible cases.

**Case 1** For  $x_{t+1} = 0$ , there exists exactly one state with a relative fitness  $f_{(x_t, x_t)} > 0$  for population of fixed size  $N$  as the state will transit to itself every time. It can be concluded that,

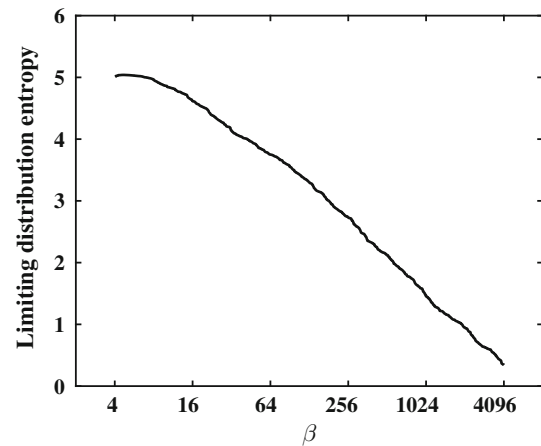
$$P_m^h(\bar{x}_t | \bar{x}_t) = p_m^{\text{poly}_j^l(x_t, x_t)} (1 - p_m)^{\text{poly}_j^l(x_t, x_t)} > 0 \quad (l, j = 0) \tag{7}$$

**Case 2** For  $x_{t+1} > 0$ ,

$$P_m^h(x_{t+1} | \bar{x}_t) = \sum_{l \in h} \left( \sum_{j \in d^l} p_m^{\text{poly}_j^l(x_{t+1}, x_t)} (1 - p_m)^{\text{poly}_j^l(x_{t+1}, x_t)} \right) > 0 \quad (l, j > 0) \tag{8}$$

Therefore, for arbitrary states  $(x_t, x_{t+1}) \in X$ , there exists strictly positive numbers in  $P_m^h$ . As both crossover and selection transitions are strictly positive, the entry in  $P_r$  consists of a product of strictly positive numbers.  $\square$

As the stationary distribution is strictly positive, all states are considered to be associated with finite probability. This means that a limiting behavior may be possible and to showcase the possibility, it is important to evaluate the entropy with respect to increase in population size inside each local block during mutation operation. The usefulness of hierarchical mutation scheme is shown in Fig. 3 by presenting the variations in limiting distribution entropy [16] with respect to  $\beta$ . The results illustrate the approximate computed stationary distribution entropy on a test problem (ZDT1) at different selected values of  $\beta$ . The values of  $\beta$  are evaluated by considering  $d = 4$  and  $1 \leq h \leq 6$ . It can be observed from Fig. 3 that the entropy reduces to zero with increase in  $\beta$ . Further, it can be



**Fig. 3** Variation of limiting distribution entropy with local block population  $\beta$

suggested that with increase in  $\beta$ , it may be possible to approach limiting distribution behavior (i.e., a globally optimal solution with probability one) by making the probability associated with nonoptimal states as small as required. Since mutation operation is carried out for both polynomial and binary coded genetic evolution in such a way to enforce the relative fitness,  $f > 0$  [19], it may be possible to parameterize  $h$ NSGA-II to reduce bias towards solutions by analyzing ergodic bound of hierarchical mutation probability. The presence of uncertainty in transitions during hierarchical mutation to choose between next state and current state can cause finite number of revisits to already traversed state showing condition of transient states  $(P_m^h(\bar{y} | \bar{x}) < \infty)$  (Theorem 2). Further, it may account for few states to be recurrent  $(P_m^h(\bar{y} | \bar{x}) = \infty)$  if the state transits to itself in all transitions. In such cases, variations in population size parameter  $\beta$  inside local block of  $h$ NSGA-II has important implications for recurrence and transience conditions.

**Proposition 2** In any state space  $X$ , a state  $x_t$  is recurrent if and only if  $\sum_m^\infty P_m^h(\bar{x}_t | \bar{x}_t) = \infty$ , transient otherwise [19].

**Theorem 2** All states in  $X$  of a Markov chain are either recurrent or all states in  $X$  are transient. Thus, if there is a transition from  $x_t$  to  $x_{t+1}$  and state  $x_t$  is assumed to be recurrent, then  $x_{t+1}$  is also recurrent. Equivalently if a transition occurs from state  $x_t$  to  $x_{t+1}$  and state  $x_t$  is transient, then state  $x_{t+1}$  is also transient. In particular, it can be said that for an irreducible Markov chain, either all states are recurrent or all states are transient [19].

**Proof** If there is a transition between  $x_t$  and  $x_{t+1}$  and  $x_t$  is recurrent, then  $x_{t+1}$  must be recurrent because every time  $x_t$  is revisited after some interval with a success probability,  $x_{t+1}$  will also be revisited with a finite time interval. As  $x_t$  is recurrent, it will be visited a number of times with a

probable success rate making  $x_{t+1}$  to be recurrent. Eventually, this can be viewed to be a sequence of random walk repeating itself at regular intervals.

### 3.4 Conditions for ergodicity

Mutation operation plays a prominent role in robust convergence of a simple genetic algorithm. With inclusion of hierarchical mutation operation, the number of individuals participating in genetic evolution process is comparatively more than NSGA-II, which in turn increases the search space of  $h$ NSGA-II and ensures a rigid schedule bound for ergodicity. In this section, we present a schedule bound for hierarchical mutation in  $h$ NSGA-II that will establish strong ergodicity. There are a number of conditions to showcase the existence of ergodicity in the proposed Markov chain. One of the necessary and sufficient conditions for ergodicity is the existence of a non-trivial measure [20],  $\mu$  satisfying (9).

$$\mu_m^h(\bar{y}) \geq \int_X \mu_m^h(d\bar{x}) P_m^h(\bar{y}|\bar{x}) \tag{9}$$

According to general state space context, it is quite difficult to verify the criterion in (9). Therefore, a Doeblin condition [21] can be established along with the condition in (9) satisfying (10) having a probability measure ( $\mu$ ) and a fixed integer  $\delta, 0 < \delta \leq \beta$ , that can account for a schedule bound for ergodicity of the proposed Markov chain model, i.e.,

$$P_m^h(\bar{y}|\bar{x}) \leq \delta, \quad \forall \bar{x} \in X \tag{10}$$

**Theorem 3** *Following both conditions in (9) and (10), a sufficient condition for the proposed Markov chain to be ergodic is the existence of a non-trivial measurable function  $g$  such that [20],*

$$\int_X P_m^h(d\bar{y}|\bar{x}) g_m^h(\bar{y}) \leq g_m^h(\bar{x}) - 1, \quad \forall \bar{x} \notin X \tag{11}$$

and for a fixed integer  $\delta > 0$ ,

$$\int_X P_m^h(d\bar{y}|\bar{x}) g_m^h(\bar{y}) = \omega(\bar{x}) \leq \delta < \infty, \quad \forall \bar{x} \in X \tag{12}$$

*Proof* Considering  $g$  to be a non-measurable function as described in Theorem 3 and let there is a transition from state  $\bar{x}$  to state  $\bar{y}$  using mutation probability,  $P_m^h$  having height  $h$  of a each local block, then at  $(t + 1)$ th time point,

$$g_{m(t+1)}^h(\bar{x}) = \int_X P_{m(t)}^h(d\bar{y}|\bar{x}) g_m^h(\bar{y}), \quad t \geq 1$$

If there exists an intermediate state space  $\kappa, \kappa^c \in X$ , such that  $X = \kappa \cup \kappa^c$  then

$$\begin{aligned} g_{m(t+2)}^h(\bar{x}) &= \int_X P_{m(t)}^h(d\bar{z}|\bar{x}) \left[ \int_X P_{m(t)}^h(d\bar{y}|\bar{z}) g_m^h(\bar{y}) \right] \\ &\leq \int_{\kappa} P_{m(t)}^h(d\bar{z}|\bar{x}) \omega(\bar{z}) + \int_{\kappa^c} P_{m(t)}^h(d\bar{z}|\bar{x}) [g_m^h(\bar{z}) - 1] \\ &\leq \int_{\kappa} P_{m(t)}^h(d\bar{z}|\bar{x}) [\omega(\bar{z}) + 1] \\ &\quad + \int_{\kappa^c} P_{m(t)}^h(d\bar{z}|\bar{x}) g_m^h(\bar{z}) - P_{m(t)}^h(X|\bar{x}) \\ &\leq (\delta + 1) P_{m(t)}^h(\kappa|\bar{x}) + g_{m(t+1)}^h(\bar{x}) - 1 \end{aligned} \tag{13}$$

Iterating (13) and dividing by  $t$ , we have

$$\begin{aligned} \frac{1}{t} g_{m(t+2)}^h(\bar{x}) &\leq (\delta + 1) \left[ \frac{1}{t} \sum_{i=1}^t P_{m(i)}^h(\kappa|\bar{x}) \right] \\ &\quad + \frac{1}{t} g_{m(2)}^h(\bar{x}) - \frac{1}{t} \sum_{i=1}^t 1 \\ &\leq (\delta + 1) \left[ \frac{1}{t} \sum_{i=1}^t P_{m(i)}^h(\kappa|\bar{x}) \right] + \frac{1}{t} g_{m(2)}^h(\bar{x}) - 1 \end{aligned} \tag{14}$$

As per [20], for  $\forall x$ ,

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^t P_{m(i)}^h(\kappa|\bar{x}) = \pi(\kappa|\bar{x}) \tag{15}$$

exists and  $\pi(\kappa|\bar{x})$  is finite when  $\kappa \in X$ . With increase in time point as  $t \rightarrow \infty$  and by the non-negativity of  $g_{m(t)}^h$  and finiteness of  $g_{m(2)}^h$  in (14), we have

$$\pi(\kappa|\bar{x}) \geq \frac{1}{\delta + 1} \geq \frac{1}{\beta} > 0 \tag{16}$$

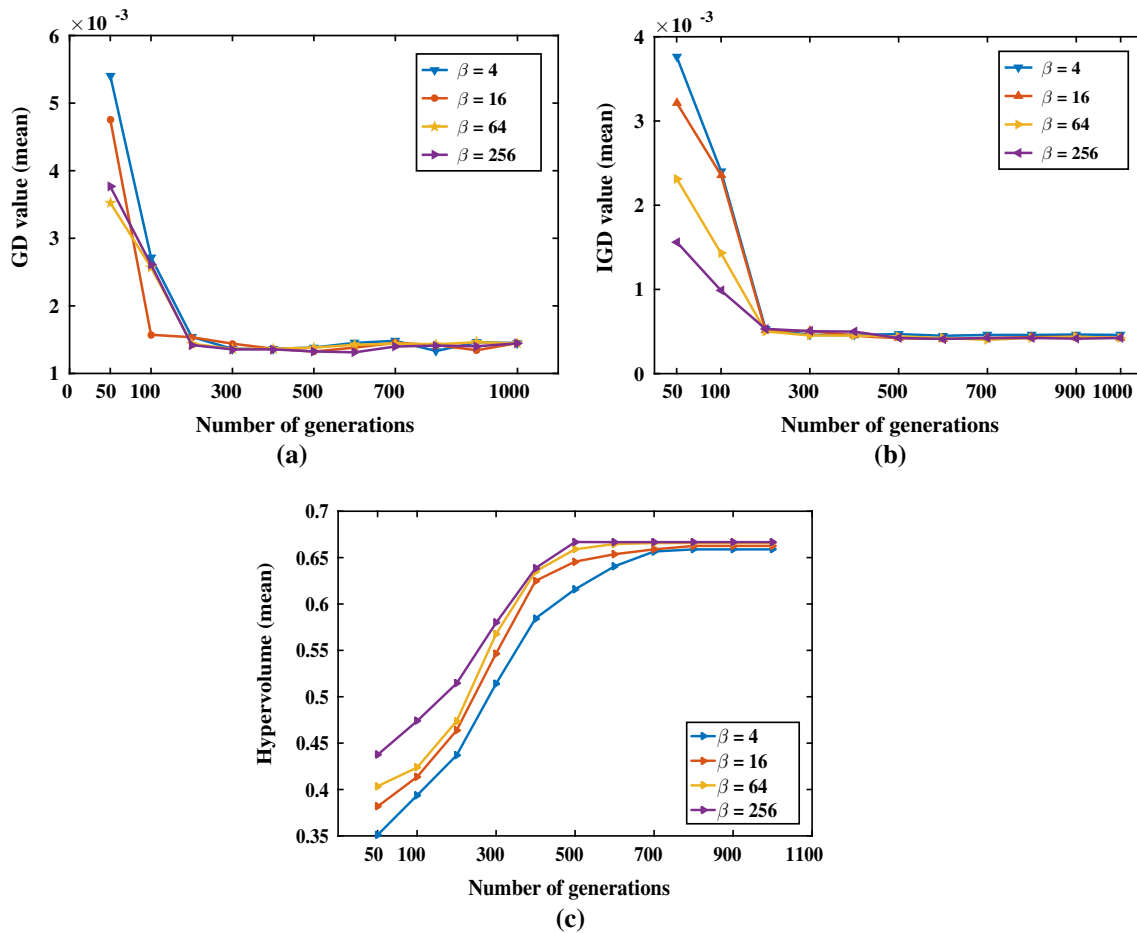
where for a finite population size  $\beta$ , there exists a schedule bound for ergodicity of proposed Markov chain model.

## 4 Performance analysis

### 4.1 Parametric study

In this subsection, we perform a parametric study for population size parameter  $\beta$  of hierarchical polynomial mutation operator. As the performance of proposed hierarchical scheme can be characterized for different population sizes of each local block ( $\beta$ ), the performance can change with change in  $\beta$ . For a constant degree  $d$ , height  $h$  can be increased to postulate variations in both convergence and diversity of Pareto optimal solutions at different values of  $\beta$  for a number of generations. To measure the performance in terms of convergence towards optimal Pareto front and diversity of solutions along Pareto front, three different quality metrics are evaluated, i.e. *Generational Distance* (GD), *Inverted Generational Distance* (IGD) and





**Fig. 4** Performance of *hNSGA-II* on ZDT1 test function at different generations for various population sizes of local block ( $\beta$ ), **a** mean GD values, **b** mean IGD values, **c** mean HV values

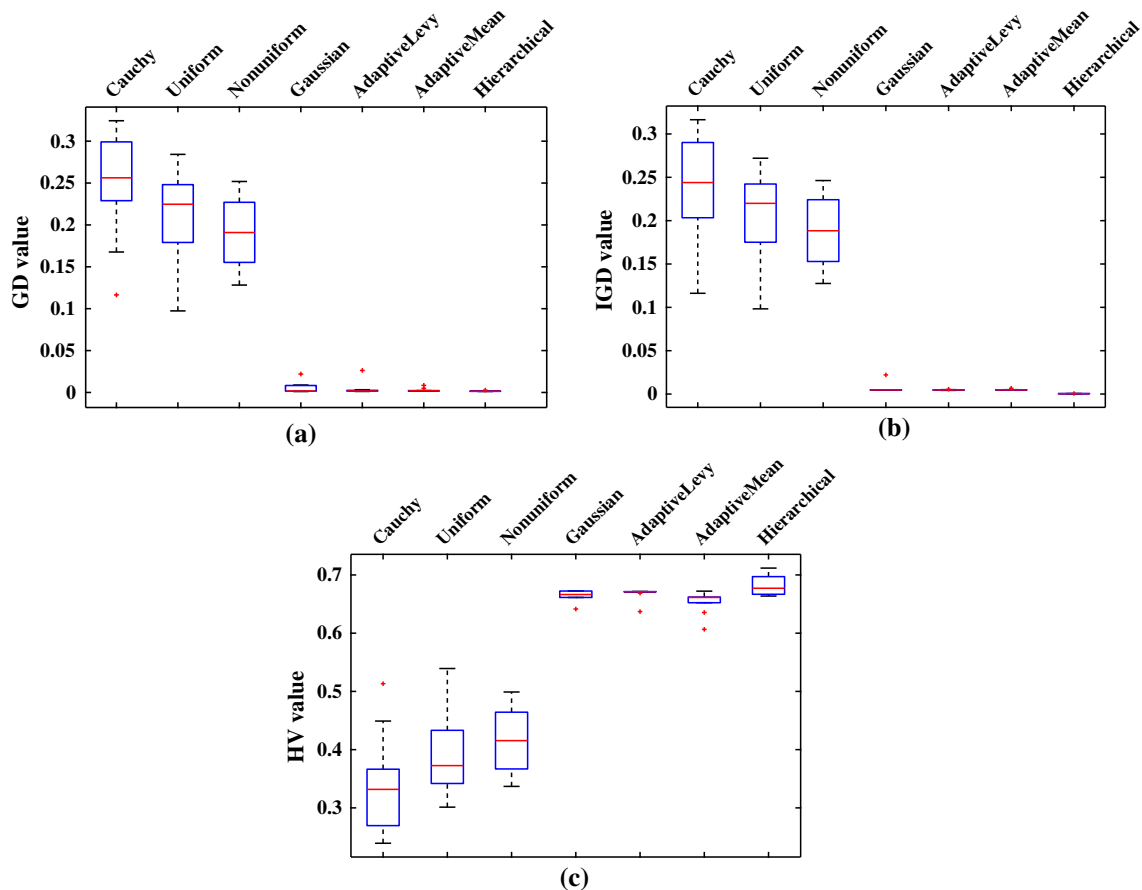
*Hypervolume* (HV). GD metric [22] is used to measure minimum Euclidean distance between generated final individuals and sample points on true Pareto front. Smaller value of GD indicates a better convergence towards true Pareto front. On the other hand, both IGD metric [22] and HV metric [23] are a measure of both convergence and diversity of solutions along true Pareto front on a single scale. If the IGD value is close to zero, it means the solutions have better convergence and diversity, and HV is a fair measure of the maximum area covered by the nondominated solutions with respect to a reference point. In view of this, several experiments are carried out on ZDT1 test function for a number of generations and it can be observed from Fig. 4 that with increase in  $\beta$ , solution (optimal PF points) attempts to attain smaller (best) GD (Fig. 4a) and IGD (Fig. 4b) values and higher HV (Fig. 4c) values at fewer generations. One of the reasons for these best values at fewer generations for higher  $\beta$  value can be due to the increase in number of offsprings in each local block with increase in  $\beta$ . It can be seen from Fig. 4(c) that with increase in number of generations, HV

values also increase. However, the HV values remain unchanged after 500 number of generations (for  $\beta = 256$  and 700 generations for other  $\beta$  values). As intermediate traces of PF exist at fewer generations, the population has to be evolved over sufficient number of generations (500 generations for  $\beta = 256$ ) to account for the convergence and diversity metric to saturate during function evaluations. With more population size, the diversity can be preserved by exploring extra search space by combining all new offsprings of each local block. Another reason for best values at fewer generations can be the reduction in selection pressure by generating more offsprings through proposed hierarchical mutation strategy. With decrease in selection pressure, the convergence issues are addressed to a great extent by mitigating the instability in crowding distance scheme. However,  $\beta$  cannot be increased to any arbitrary value. As  $\beta$  accounts for the variations in limiting distribution entropy of mutation operation, it should be kept within finite limits ( $0 < \beta < N$ ) for better performance of *hNSGA-II* (as described in Sect. 3.4).

## 4.2 Comparison with other mutation strategies

As mutation operator operates on an individual independent of other population members, it plays an important role in making overall search efficient [24]. Therefore, it is necessary to investigate the effectiveness of proposed hierarchical mutation operator by comparing the performance with other mutation operators. Hierarchical scheme for six different mutation operators (Adaptive Levy mutation [25], Cauchy mutation [26, 27], uniform mutation [28], nonuniform mutation [28], Gaussian mutation [29] and Adaptive-mean mutation [26]) are implemented and also incorporated to study the search efficiency of *h*NSGA-II framework. This study is extensive in evaluating the performance of proposed hierarchical polynomial mutation operator. Further, a total 30 decision variables ( $p = 30$ ) are considered during evaluation and each variable is bounded within  $x_i \in [0, 1]$ , where  $1 \leq i \leq 30$ . Crossover probability is kept at 0.9 and mutation probability is set to  $1/p$  for all mutation schemes. Both crossover and mutation distribution indices are set to 20 during the entire process. As various mutation operators are employed within *h*NSGA-II

framework, different parameters are addressed each time a mutation operator is exercised. During Cauchy mutation, a Cauchy density function [27],  $f_{cauchy} = \frac{1}{\pi} \frac{t}{t^2 + x^2}$ , where scale parameter  $t$  is set to 1 during operation. Adaptive Levy mutation scheme is employed by using parameter  $\alpha$  fixed at 1.7 for all experiments. Box-Muller transform is adopted to generate Gaussian distribution from uniformly distributed numbers during Gaussian mutation operation. Population size of each local block,  $\beta$  is kept at 256 for achieving hierarchy (as described in Sect. 4.1) during all operations. All mutation operators exercised using hierarchical scheme are compared based on convergence and diversity of solutions (mean GD values, mean IGD values and mean HV values), while 200 population individuals are allowed to evolve over 100 generations on ZDT1 test function. For each mutation operation, 30 independent runs are performed and the results are shown in Fig. 5. It can be observed from Fig. 5(a, b) that proposed hierarchical polynomial mutation operator performs better than other mutation operators as corresponding GD and IGD values are minimum. Further, it is observed from Fig. 5(c) that the proposed hierarchical polynomial mutation operator has



**Fig. 5** Performance analysis of several mutation strategies on ZDT1 test function showcasing mean, **a** GD values, **b** IGD values, **c** HV values at different generations (100–1000)

larger HV value as compared to other mutation operators. As the proposed hierarchical scheme is performed on top of polynomial mutation operator, the inherent probability distribution of creating an offspring population is similar to polynomial mutation operator, which is instrumental in the convergence of Pareto optimal solutions. Moreover, the implicit generation of distinct fitness values in each generation favors the preservation of diversity among Pareto optimal solutions.

### 4.3 Performance of *h*NSGA-II on multi-objective test functions

In this subsection, different sets of multi-objective test functions are solved using hierarchical polynomial mutation based *h*NSGA-II framework. These test functions have been used as standard benchmarks in literature. To demonstrate the applicability of *h*NSGA-II, a total of twelve test problems with two different groups of benchmarks are considered (both constrained and unconstrained test problems). With the aim of giving a complete overview of performance of *h*NSGA-II, several standard optimization algorithms (NSGA-II-DE [30], MOEA/D-DE [30], NSGA-II [13],  $\epsilon$ -MOEA [17], NSGA-IIa [31] and ssNSGA-II [31]) are considered for evaluation on all unconstrained benchmarks (ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, Schaffer, Fonseca, and Kursawe test functions [13]). For the purpose of comparison, we have used the authors implementation of all algorithms. However, the implementation framework of NSGA-II-DE and MOEA/D-DE does not have any provision of constraint handling management. Therefore, during showcase of results, we are forced to leave the spaces blank during the analysis of constrained benchmarks (Binh2, ConstrEx, Ozyczka2 and Tanaka test functions [13]). During evaluation of all algorithms, a population size of 200 is considered to evolve over 500 generations. For all NSGA-II and MOEA variants (i.e., NSGA-II, NSGA-IIa, ssNSGA-II, NSGA-II-DE, MOEA/D-DE and  $\epsilon$ -MOEA) and proposed *h*NSGA-II approach, the crossover probability is kept at 0.9. Mutation probability is set to  $1/p$  for all algorithms, where  $p$  is the number of decision variables. The crossover and mutation distribution indices are set to 20 as commonly used in literature. The population size parameter ( $\beta$ ) of each local block is set to 256 to have a reasonable amount of offspring population to preserve sufficient diversity across the obtained PF. Table 1 represents the mean and standard deviations of GD values for *h*NSGA-II along with other standard optimization algorithms based on 30 independent runs. It can be seen from Table 1 that GD values are better for seven out of twelve benchmarks (ZDT1, ZDT2, ZDT3, Fonseca, Kursawe, Ozyczka2 and Tanaka [13]), while the performance is inferior for both ZDT4, ZDT6, Schaffer,

Binh2 and ConstrEx benchmarks. This indicates that the resulting Pareto fronts from *h*NSGA-II are closer to the optimal Pareto fronts as compared to those computed by other standard optimization algorithms. From Table 2, it can be seen that *h*NSGA-II obtains lowest (best) values for the IGD metric in seven out of twelve benchmarks (ZDT1, ZDT2, ZDT3, Schaffer, Fonseca, Kursawe and Tanaka). This points out that the obtained Pareto fronts of multi-objective test problems are closer to true Pareto fronts and the non-dominated solutions are better spread for *h*NSGA-II than other algorithms. Again to showcase the performance of *h*NSGA-II as a measure of both convergence and diversity in a concise manner, HV metric is evaluated for all benchmarks (as described in Sect. 4.1). It can be seen from Table 3 that *h*NSGA-II obtains best (highest) HV values in six out of twelve benchmarks. In view of the results, *h*NSGA-II shows better performance than other algorithms (seven out of twelve, seven out of twelve, and six out of twelve for GD, IGD and HV metric respectively). The usage of hierarchical scheme in polynomial mutation to generate an elite offspring appears to be fruitful in order to achieve better convergence and diversity of solutions along true PF. However, the reason for better performance can be attributed to the generation of individuals having unique fitness values during mutation operation of *h*NSGA-II.

## 5 Application to analog/RF circuit optimization

At the lowest level of design hierarchy, the process of circuit synthesis comes down to the step of circuit optimization that makes use of sizing principle and biasing performances of all devices in the circuit to meet certain design constraints. Once an appropriate topology is selected during design process, certain performance specifications are represented as objective functions by formulating a set of physical equations that relate the device characteristics to design parameters. However, it is difficult to solve these equations explicitly under a set of design constraints. Therefore, it is necessary to employ numerical optimization techniques to implicitly analyze these equations while optimizing the circuit under certain design constraints.

The design of analog and RF circuits has become complex with the advent of deep submicron technology. It has imposed a discontinuity on the seer practice of knowledge-based sizing [32] which has been a popular choice of design methodology since the beginning of semiconductor design and manufacturing. However, with increase in integration density and complexity, the use of simple models designed using this methodology will lead

**Table 1** The GD statistics (mean and SD values) based on 30 independent runs of *h*NSGA-II for different multi-objective test problems keeping  $\beta = 256$ 

| GD metric |            |                   |                   |                   |            |                   |                   |
|-----------|------------|-------------------|-------------------|-------------------|------------|-------------------|-------------------|
| Problem   | NSGA-II-DE | MOEA/D-DE         | NSGA-II           | $\epsilon$ -MOEA  | NSGA-IIa   | ssNSGA-II         | <i>h</i> NSGA-II  |
| ZDT1      |            |                   |                   |                   |            |                   |                   |
| Mean      | 2.3897e−03 | 3.2241e−03        | 1.1517e−01        | 1.5656e−03        | 1.8665e−01 | 8.2189e−01        | <b>1.4490e−03</b> |
| SD        | 8.2341e−04 | <b>1.2564e−05</b> | 4.5101e−03        | 1.2950e−04        | 1.0490e−03 | 4.9559e−03        | 5.0660e−05        |
| ZDT2      |            |                   |                   |                   |            |                   |                   |
| Mean      | 2.0976e−01 | 2.7419e−01        | 2.0507e−01        | 2.5425e−03        | 1.4225e−01 | 0.1250e+01        | <b>2.3122e−03</b> |
| SD        | 3.7761e−04 | 4.8112e−03        | 2.1270e−03        | 0.1812e−03        | 1.5926e−03 | 9.0625e−01        | <b>3.1245e−05</b> |
| ZDT3      |            |                   |                   |                   |            |                   |                   |
| Mean      | 2.3542e−01 | 1.7304e−01        | 9.6721e−02        | 1.7512e−03        | 2.1417e−01 | 3.5356e−01        | <b>1.6974e−03</b> |
| SD        | 6.2060e−03 | 1.9120e−03        | 3.0611e−03        | 0.0512e−03        | 2.3206e−02 | 2.4369e−02        | <b>5.3871e−05</b> |
| ZDT4      |            |                   |                   |                   |            |                   |                   |
| Mean      | 7.2736e−02 | <b>1.7458e−03</b> | 4.0117e+01        | 5.9214e−03        | 1.6630e−01 | 3.3299e+01        | 0.2816e+01        |
| SD        | 4.1730e−04 | <b>3.0129e−05</b> | 7.0126e−02        | 2.2344e−04        | 2.8151e−03 | 6.6099e−02        | 3.1011e−02        |
| ZDT6      |            |                   |                   |                   |            |                   |                   |
| Mean      | 2.3307e−01 | 2.8336e−01        | 0.2416e+01        | <b>5.9950e−03</b> | 1.8020e−01 | 0.2981e+01        | 2.0486e−01        |
| SD        | 7.0183e−02 | 7.2300e−02        | 4.0181e−01        | <b>1.1209e−03</b> | 1.3987e−02 | 2.0804e−02        | 4.6200e−03        |
| Schaffer  |            |                   |                   |                   |            |                   |                   |
| Mean      | 8.4757e−01 | 8.5561e−01        | 1.1460e−02        | <b>9.0352e−03</b> | 1.1128e−02 | 4.1867e+02        | 1.0514e−02        |
| SD        | 1.0070e−03 | 1.1200e−03        | 1.0123e−03        | <b>1.2420e−04</b> | 2.2706e−03 | 0.4116e−01        | 2.2311e−04        |
| Fonseca   |            |                   |                   |                   |            |                   |                   |
| Mean      | 2.0977e−01 | 2.7198e−01        | 2.4156e−03        | 5.3019e−03        | 1.1666e−01 | 5.3519e−02        | <b>1.2167e−03</b> |
| SD        | 2.1231e−03 | 3.4230e−03        | 1.5621e−03        | 2.3120e−03        | 3.1184e−02 | 3.9624e−03        | <b>6.7810e−04</b> |
| Kursawe   |            |                   |                   |                   |            |                   |                   |
| Mean      | 1.7926e+01 | 1.7886e+01        | 7.3262e−03        | 1.0987e−02        | 6.8233e−01 | 7.9358e−03        | <b>4.3495e−03</b> |
| SD        | 4.6431e−01 | 2.4451e−01        | 2.3113e−04        | 3.9841e−03        | 3.3847e−02 | <b>1.6395e−04</b> | 4.0192e−04        |
| Binh2     |            |                   |                   |                   |            |                   |                   |
| Mean      | –          | –                 | 1.8161e+01        | 0.6896e+01        | 1.6991e−01 | <b>9.5720e−03</b> | 1.9121e−01        |
| SD        | –          | –                 | 2.3001e−01        | 3.9841e−03        | 3.3847e−02 | <b>1.6395e−04</b> | 4.0192e−04        |
| ConstrEx  |            |                   |                   |                   |            |                   |                   |
| Mean      | –          | –                 | 6.0349e−03        | <b>1.4202e−03</b> | 2.8380e−01 | 1.7491e−03        | 7.2328e−03        |
| SD        | –          | –                 | <b>1.5887e−05</b> | 5.3812e−05        | 2.5671e−03 | 5.0045e−05        | 1.7242e−05        |
| Ozyczka2  |            |                   |                   |                   |            |                   |                   |
| Mean      | –          | –                 | 0.2725e+01        | 1.6004e+00        | 5.0831e−02 | 1.8653e−01        | <b>2.2431e−02</b> |
| SD        | –          | –                 | 1.7752e−01        | 0.3041e−01        | 2.8847e−04 | 3.3040e−03        | <b>2.1790e−04</b> |
| Tanaka    |            |                   |                   |                   |            |                   |                   |
| Mean      | –          | –                 | 1.1905e−02        | 1.7084e−02        | 9.3017e−02 | 1.7205e−02        | <b>8.1449e−03</b> |
| SD        | –          | –                 | 4.6172e−04        | 2.1430e−04        | 4.7349e−04 | 2.3310e−04        | <b>3.1295e−05</b> |

The results in boldface indicate the best values

to false design. Therefore, new approaches [33, 34] are presented in literature to handle the design complexity of integrated circuit chip. These approaches appear to translate the problem of circuit design into a function minimization or maximization problem, which can be solved through numerical techniques [33]. Mostly, it starts with the development of an efficient performance evaluator

framework, which evaluates the minimization (maximization) function [33]. This function is created based on the behavior of circuit topology [33, 35, 36] and they rely on the analytical equation formulation. The efficiency of the evaluator lies in the efficient design of circuit equations. Although a number of deterministic algorithms have been presented to perform circuit design, the assessment of

**Table 2** The IGD statistics (mean and SD values) based on 30 independent runs of *h*NSGA-II for different multi-objective test problems keeping  $\beta = 256$ 

| IGD Metric |            |                   |                   |                   |            |                   |                   |
|------------|------------|-------------------|-------------------|-------------------|------------|-------------------|-------------------|
| Problem    | NSGA-II-DE | MOEA/D-DE         | NSGA-II           | $\epsilon$ -MOEA  | NSGA-IIa   | ssNSGA-II         | <i>h</i> NSGA-II  |
| ZDT1       |            |                   |                   |                   |            |                   |                   |
| Mean       | 1.8796e−03 | 5.4652e−03        | 7.9356e−02        | 4.5175e−03        | 1.7856e−02 | 5.9368e−02        | <b>5.3687e−04</b> |
| SD         | 2.3201e−05 | 3.2131e−05        | 1.2910e−03        | 1.1810e−04        | 1.8179e−03 | 6.9291e−03        | <b>1.4121e−05</b> |
| ZDT2       |            |                   |                   |                   |            |                   |                   |
| Mean       | 1.6733e−01 | 2.2529e−01        | 1.4510e−01        | 4.3611e−03        | 1.1558e−02 | 8.8897e−01        | <b>2.3605e−03</b> |
| SD         | 0.1214e−01 | 2.0805e−02        | 2.4139e−03        | 7.0231e−03        | 3.1240e−03 | 4.1001e−02        | <b>1.0012e−04</b> |
| ZDT3       |            |                   |                   |                   |            |                   |                   |
| Mean       | 2.4908e−01 | 2.1255e−01        | 7.2018e−03        | 1.2465e−02        | 1.8420e−02 | 3.7037e−02        | <b>8.0021e−04</b> |
| SD         | 1.5988e−02 | 3.1282e−02        | 2.7836e−04        | 2.4660e−03        | 2.1680e−03 | 1.0476e−03        | <b>1.0798e−05</b> |
| ZDT4       |            |                   |                   |                   |            |                   |                   |
| Mean       | 5.7595e−02 | <b>1.2372e−03</b> | 1.4407e+01        | 1.4264e−02        | 1.8139e−02 | 0.2155e+01        | 0.2497e+01        |
| SD         | 1.2400e−03 | <b>2.0912e−04</b> | 2.1740e+00        | 5.1265e−03        | 1.3541e−02 | 1.5251e−01        | 1.3100e+00        |
| ZDT6       |            |                   |                   |                   |            |                   |                   |
| Mean       | 1.8216e−01 | 2.5408e−01        | 8.4401e−01        | <b>1.6614e−02</b> | 1.3146e−02 | 2.2793e−01        | 2.2087e−02        |
| SD         | 5.6002e−02 | 1.4320e−02        | 1.4013e−02        | 1.8120e−03        | 2.1457e−03 | 1.2841e−02        | <b>1.0756e−03</b> |
| Schaffer   |            |                   |                   |                   |            |                   |                   |
| Mean       | 0.1467e+01 | 0.2560e+01        | 2.2365e−03        | 3.2884e−02        | 1.8185e−02 | 2.9532e+01        | <b>2.0502e−03</b> |
| SD         | 0.2003e+00 | 0.4109e+00        | 4.8701e−05        | 2.3212e−05        | 6.6150e−03 | 0.6603e+00        | <b>1.0028e−05</b> |
| Fonseca    |            |                   |                   |                   |            |                   |                   |
| Mean       | 1.2627e−01 | 1.7374e−01        | 7.8359e−04        | 1.3212e−02        | 2.1680e−02 | 8.3367e−03        | <b>5.3678e−04</b> |
| SD         | 7.9835e−02 | 4.4459e−02        | <b>2.0024e−05</b> | 2.6004e−03        | 3.2780e−03 | 1.0258e−04        | 3.0901e−05        |
| Kursawe    |            |                   |                   |                   |            |                   |                   |
| Mean       | 1.8181e+01 | 1.829e+01         | 1.2120e−02        | 8.7333e−02        | 1.0476e−01 | 6.0136e−03        | <b>5.1619e−03</b> |
| SD         | 1.2530e+00 | 1.3156e+00        | 2.1240e−03        | 2.1756e−03        | 1.6538e−02 | <b>1.5224e−05</b> | 1.0967e−04        |
| Binh2      |            |                   |                   |                   |            |                   |                   |
| Mean       | –          | –                 | 3.8746e−02        | 1.5039e−01        | 1.3822e−02 | <b>9.9204e−05</b> | 4.3515e−02        |
| SD         | –          | –                 | 3.4132e−04        | 2.7792e−03        | 4.8685e−04 | <b>1.9055e−07</b> | 1.1563e−04        |
| ConstrEx   |            |                   |                   |                   |            |                   |                   |
| Mean       | –          | –                 | 2.6953e−03        | 1.0047e−02        | 1.3330e−02 | <b>1.7491e−03</b> | 2.4158e−03        |
| SD         | –          | –                 | 4.01551e−05       | 4.0237e−04        | 1.5281e−04 | 1.3506e−05        | <b>1.0328e−05</b> |
| Ozyczka2   |            |                   |                   |                   |            |                   |                   |
| Mean       | –          | –                 | 6.1283e−01        | 3.7564e+00        | 4.3907e−02 | <b>1.4653e−02</b> | 2.1583e−01        |
| SD         | –          | –                 | 8.9101e−03        | 4.7949e−01        | 5.3761e−04 | 4.5504e−03        | <b>1.3102e−03</b> |
| Tanaka     |            |                   |                   |                   |            |                   |                   |
| Mean       | –          | –                 | 1.3635e−03        | 4.9998e−03        | 2.5803e−02 | 5.1985e−03        | <b>9.5842e−04</b> |
| SD         | –          | –                 | 7.7346e−05        | 2.6201e−05        | 6.0021e−03 | 2.3179e−05        | <b>1.3069e−06</b> |

The results in boldface indicate the best values

initial design point requires prior knowledge or additional design equation formulation. This favours the usage of indeterministic approaches, like use of nature-inspired algorithms in circuit design. A number of efforts on nature-inspired algorithms have been made in literature to capture the design space boundaries of analog and RF circuits. Genetic algorithm (GA) is employed as one of the popular

optimization routines for optimization of analog and RF circuits in both industry and academia [4, 37]. Several variants of GAs are also being employed to design CMOS operational amplifier circuits with constrained topologies. Genetic programming is used as an analog synthesis tool in several filter designs and circuit synthesis problems [4]. With the advent of new and complex circuits, it is

**Table 3** The HV statistics (mean and SD values) based on 30 independent runs of *h*NSGA-II for different multi-objective test problems keeping  $\beta = 256$ 

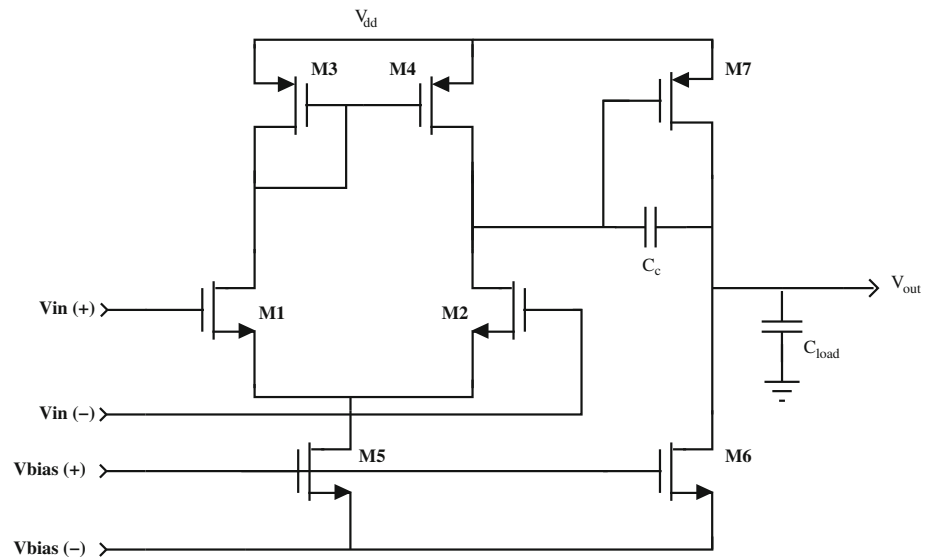
| HV Metric |                |                |         |                  |                |           |                  |
|-----------|----------------|----------------|---------|------------------|----------------|-----------|------------------|
| Problem   | NSGA-II-DE     | MOEA/D-DE      | NSGA-II | $\epsilon$ -MOEA | NSGA-IIa       | ssNSGA-II | <i>h</i> NSGA-II |
| ZDT1      |                |                |         |                  |                |           |                  |
| Mean      | 0.63629        | 0.66644        | 0.65754 | 0.56680          | 0.06493        | 0.05936   | <b>0.66661</b>   |
| SD        | 0.01321        | 0.01421        | 0.01214 | 0.01518          | 0.00032        | 0.00021   | <b>0.03309</b>   |
| ZDT2      |                |                |         |                  |                |           |                  |
| Mean      | 0.40421        | 0.33118        | 0.32413 | 0.32840          | 0.07865        | 0.04019   | <b>0.43185</b>   |
| SD        | 0.00214        | 0.00534        | 0.00231 | 0.00491          | 0.00011        | 0.00009   | <b>0.01098</b>   |
| ZDT3      |                |                |         |                  |                |           |                  |
| Mean      | 0.48564        | 0.51609        | 0.51313 | 0.70033          | 0.00696        | 0.002151  | <b>0.78175</b>   |
| SD        | 0.00207        | 0.01082        | 0.01202 | 0.03689          | 1.964e−05      | 5.093e−05 | <b>0.04129</b>   |
| ZDT4      |                |                |         |                  |                |           |                  |
| Mean      | 0.46089        | <b>0.66464</b> | 0.65182 | 0.48224          | 0.07544        | 0.15132   | 0.64885          |
| SD        | 0.00403        | <b>0.09049</b> | 0.07091 | 0.00510          | 0.000781       | 0.001051  | 0.03092          |
| ZDT6      |                |                |         |                  |                |           |                  |
| Mean      | <b>0.46790</b> | 0.40472        | 0.37734 | 0.21348          | 0.13621        | 0.18285   | 0.21876          |
| SD        | <b>0.00602</b> | 0.004192       | 0.00238 | 0.00157          | 0.00045        | 0.00084   | 0.05603          |
| Schaffer  |                |                |         |                  |                |           |                  |
| Mean      | 0.63977        | <b>0.79363</b> | 0.66207 | 0.61400          | 0.78868        | 0.35681   | 0.66182          |
| SD        | 0.04873        | 0.05092        | 0.04704 | 0.03546          | <b>0.05241</b> | 0.00310   | 0.03812          |
| Fonseca   |                |                |         |                  |                |           |                  |
| Mean      | <b>0.40921</b> | 0.31551        | 0.30796 | 0.27937          | 0.06522        | 0.15471   | 0.30507          |
| SD        | 0.00319        | 0.00592        | 0.00531 | 0.00462          | 0.00013        | 0.00025   | <b>0.00901</b>   |
| Kursawe   |                |                |         |                  |                |           |                  |
| Mean      | 0.42537        | 0.43047        | 0.39946 | 0.25383          | 0.03120        | 0.16538   | <b>0.49325</b>   |
| SD        | 0.00340        | 0.00400        | 0.00143 | 0.00487          | 0.00040        | 0.00053   | <b>0.00561</b>   |
| Binh2     |                |                |         |                  |                |           |                  |
| Mean      | –              | –              | 0.65098 | 0.64326          | <b>0.99475</b> | 0.72786   | 0.75096          |
| SD        | –              | –              | 0.05289 | 0.06164          | <b>0.15123</b> | 0.08931   | 0.09231          |
| ConstrEx  |                |                |         |                  |                |           |                  |
| Mean      | –              | –              | 0.63811 | 0.70660          | <b>0.99030</b> | 0.72916   | 0.73804          |
| SD        | –              | –              | 0.05037 | 0.03033          | <b>0.17546</b> | 0.05861   | 0.07500          |
| Ozyczka2  |                |                |         |                  |                |           |                  |
| Mean      | –              | –              | 0.36205 | 0.43741          | 0.0            | 0.27652   | <b>0.45255</b>   |
| SD        | –              | –              | 0.00246 | 0.00104          | 0.0            | 0.00093   | <b>0.00269</b>   |
| Tanaka    |                |                |         |                  |                |           |                  |
| Mean      | –              | –              | 0.33427 | 0.62590          | 0.11264        | 0.23295   | <b>0.65115</b>   |
| SD        | –              | –              | 0.00135 | 0.01902          | 0.00062        | 0.00330   | <b>0.03962</b>   |

The results in boldface indicate the best values

necessary to come up with new design strategies using efficient techniques. Therefore, many nature-inspired algorithms like, particle swarm optimization (PSO) [6], simulated annealing (SA) [36], ant colony optimization (ACO) [38], etc. are also suggested as one of the design space boundary exploration techniques for circuit optimization. These techniques are employed to design efficient recursive digital filters with infinite impulse response

[39] and for finite impulse response (FIR) filters [40, 41]. Besides these research efforts, several commercial design tools have also been introduced for circuit sizing in the past few years, such as NeoCircuit [42], Genius's ADA [43], Barcelona Design [44], etc. However, these techniques are employed to optimize a single cost function instead of an optimally located space spanned by a number of cost functions (multi-objective optimization). As circuit

**Fig. 6** Two-stage operational amplifier circuit [48]



modeling is often affected by uncertainties and errors at many levels, it is necessary to analyze the trade-offs among performance specifications to avoid any measurement errors and presence of inconsistent solutions during design. To overcome these problems, several multi-objective techniques are presented in literature with emphasis on nature-inspired algorithms. A hybrid combination of SA and PSO is also suggested in [45] to find solutions to cope with design specifications at circuit-level and exploration of trade-offs of different analog circuits. Among all presented multi-objective techniques, NSGA-II [13] is employed as a leading optimizer in various analog and RF IC synthesis tools [46, 47].

In this section, the proposed *h*NSGA-II method is employed to optimize the performance of circuit. All circuit design parameters are given as input bounds to optimize two contradicting design costs which are realized by evaluating a Pareto front. Here, an analog circuit (two-stage operational transconductance amplifier) and an RF circuit (low noise amplifier) are taken as case studies to showcase the performance and applicability of proposed *h*NSGA-II method by optimizing different design costs subject to the set of design constraints.

**Table 5** Two-stage operational amplifier constraints and specifications

| Specifications      | Description   | Constraints | Results |
|---------------------|---|-------------|---------|
| $\phi$ ( $^\circ$ ) | Phase margin  | $\geq 60$   | 60      |
| $P_{max}$ (mW)      | Max. power consumption  | $\leq 300$  | 256     |
| $SR$ (A/ $\mu$ F)   | Slew rate   | $\geq 20$   | 20      |
| $ICMR$ (V)          | Input common mode range   | [0.8, 1.6]  | 0.8     |
| $GBW$ (MHz)         | Gain bandwidth product  | $\leq 30$   | 29.5    |
| $Y$ (mV)            | $V_{in}^{min} - V_{ss} - \sqrt{\frac{I_s}{\beta}} - V_{T1}^{max}$ | $\geq 100$  | 100     |

### 5.1 Case study 1: two-stage operational amplifier

Operational amplifiers are depicted as one of the fundamental components of analog circuit due to flexible controllability. A case study of a two stage operational amplifier has been considered as shown in Fig. 6 to showcase the effectiveness of proposed *h*NSGA-II method. The design problem is formulated as a multi-objective optimization problem to analyze tradeoff between maximum gain and minimum power consumption. The closed-form expressions are approximated according to the design reported in [48]. Subsequently, the details on design

**Table 4** Two-stage operational amplifier variables and ranges

| Variables             | Description                 | Lower bound | Upper bound |
|-----------------------|-----------------------------|-------------|-------------|
| $W_1, W_2$ ( $\mu$ m) | Width of M1 and M2          | 0.48        | 0.9         |
| $W_3, W_4$ ( $\mu$ m) | Width of M3 and M4          | 2.11        | 2.7         |
| $W_6$ ( $\mu$ m)      | Width of M6                 | 30.1        | 38.88       |
| $I_{d6}$ ( $\mu$ A)   | Forward drain current of M6 | 16          | 20          |

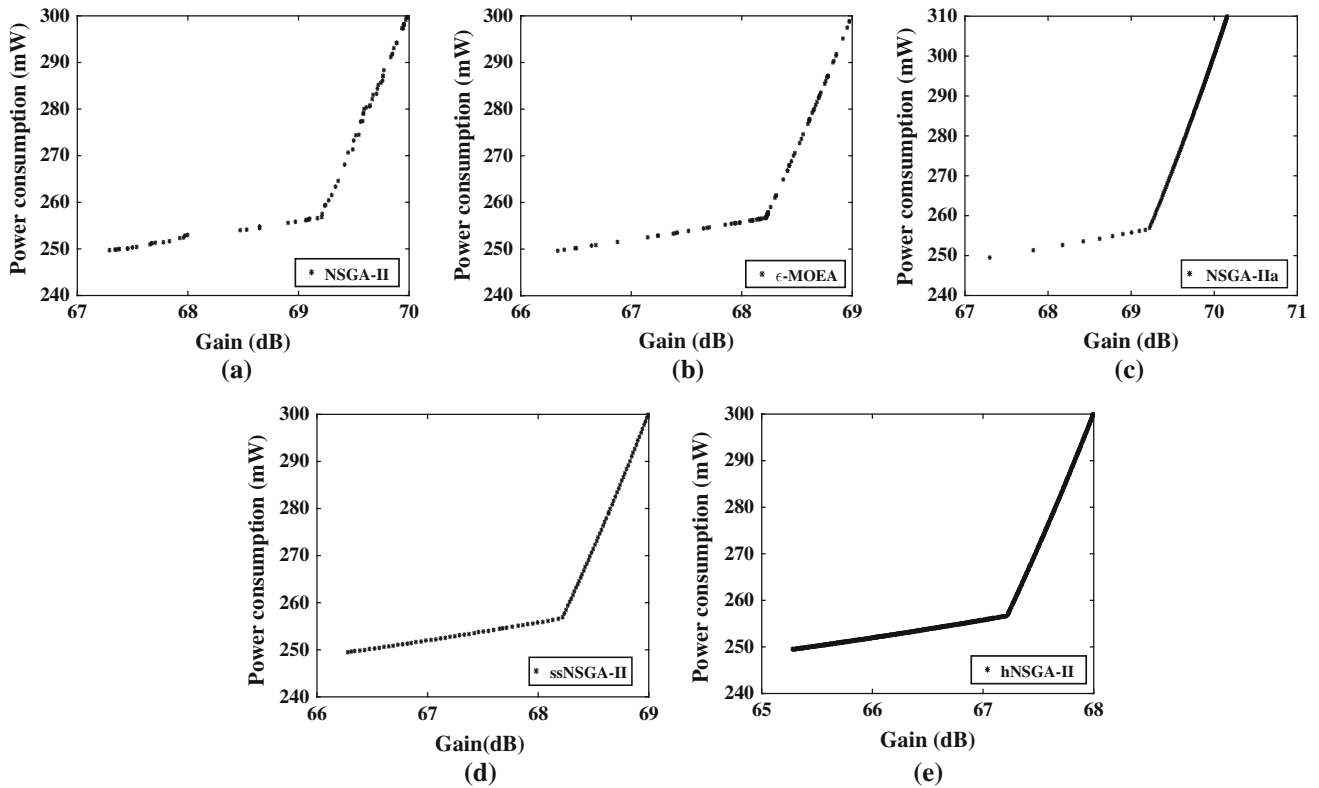


Fig. 7 Plot of nondominated fronts for gain (max) and power consumption (min) of two-stage operational transconductance amplifier by a NSGA-II, b  $\epsilon$ -MOEA, c NSGA-IIa, d ssNSGA-II, e hNSGA-II

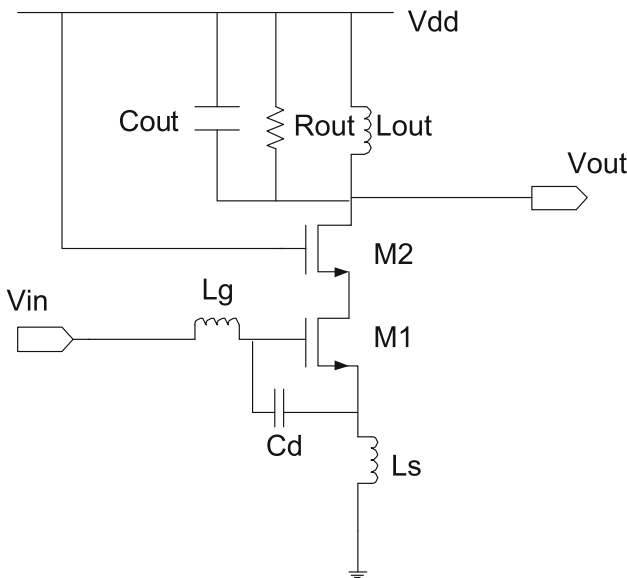


Fig. 8 Low noise amplifier circuit [50]

variables are listed in Table 4 and all design constraints are summarized in Table 5. The supply voltage is kept constant at 1.8 V for suitable operation of transistors. Before starting the process of optimization, channel lengths of 0.18  $\mu\text{m}$  are

Table 6 LNA variables and ranges

| Variables                    | Description           | Lower bound | Upper bound |
|------------------------------|-----------------------|-------------|-------------|
| $W_1, W_2$ ( $\mu\text{m}$ ) | Width of M1 and M2    | 1           | 100         |
| $I_d$ (mA)                   | Forward drain current | 0.1         | 4.5         |

selected and inversion coefficient is kept at below unity ( $\approx 0.8$ ) for M1 and M2 transistors.

However, other transistors are allowed to operate at the strong inversion side to minimize their  $V_{DS}^{\text{sat}}$  or drain thermal-noise current inputs. This suggests that the gain of the amplifier can be maximized at an expense of power consumption. During optimization of two-stage operational amplifier, 200 populations are allowed to evolve over 500 generations till all design constraints are met for efficient realization of proposed approach. Final values of all constraints after optimization of gain and power of the amplifier are listed in Table 5. Figure 7 shows the distribution of final nondominated solutions obtained by several optimization techniques including proposed hNSGA-II approach for achieving maximum gain and minimum power consumption of two-stage operational amplifier. It can be concluded from the



**Table 7** LNA constraints and specifications

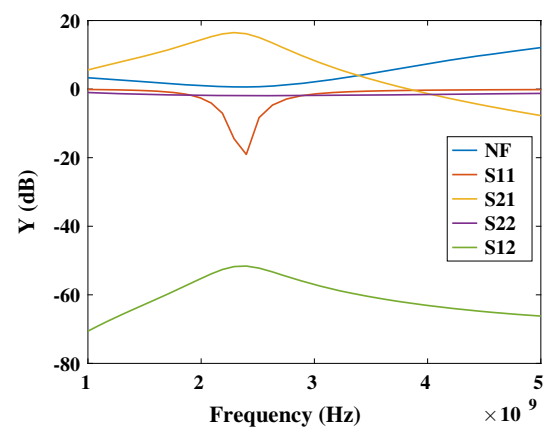
| Specifications               | Description                             | Constraints | Results |
|------------------------------|---|-------------|---------|
| $S_{11}$ (dB)                | Input reflection coefficient @ 2.4 GHz  | $\leq -20$  | - 20.02 |
| $S_{12}$ (dB)                | Reverse isolation @ 2.4 GHz             | $\leq -50$  | - 53.22 |
| $S_{21}$ (dB)                | Forward power gain @ 2.4 GHz            | $\geq 10$   | 13.01   |
| $S_{22}$ (dB)                | Output reflection coefficient @ 2.4 GHz | $\leq -0.9$ | - 1.88  |
| $NF_{min}$ (dB)              | Min. noise figure @ 2.4 GHz             | $\leq 2.5$  | 0.67    |
| $P_{max}$ (mW)               | Max. power consumption @ 2.4 GHz        | $\leq 8$    | 7.87    |
| $f_{cf}$ (GHz)               | Center frequency                        | [2.1, 2.7]  | 2.4     |
| $C_{gs}/C_{tot}$             | Capacitance ratio <sup>a</sup>          | $\leq 1$    | 0.92    |
| $Q$                          | Quality factor                          | [3, 5]      | 3.5     |
| $Z_{out}^{min}$ ( $\Omega$ ) | Min. output impedance (real part)       | $\geq 50$   | 500     |

<sup>a</sup>Ratio of intrinsic gate capacitance of  $M1$  to total capacitance ( $C_{tot} = C_{gs} + C_{ext} + C_p$ )

distribution of solutions that NSGA-II with hierarchical mutation strategy can produce better approximations of final nondominated solutions over a finite search space than other optimization techniques. Further, it can be seen from Fig. 7(e) that using  $h$ NSGA-II approach, voltage gain maximizes sub-linearly for a fixed biasing at an expense of core 256 mW power consumption under several design constraints (see Table 5). This sub-linear plot of Pareto front (PF) can be distinguished by two regions. In the first region, where gain is less than 67.5 dB, gain increases with an increase in power consumption at a slower pace. In second region, power consumption increases significantly with a small increase in gain. However, it is not desirable to achieve such higher gain at an expense of significant power consumption.

### 5.2 Case study 2: low noise amplifier (LNA)

The design of LNA is a relatively complex process as there exists tradeoffs among different performance specifications, such as noise figure (NF), transducer power gain, power consumption, area etc. Here, a case study of source degenerate cascode LNA is taken as shown in Fig. 8 to demonstrate the applicability of proposed  $h$ NSGA-II method. LNA is designed in 0.18 $\mu$ m CMOS technology with 1.8 V power supply centered around 2.4 GHz frequency using Cadence Virtuoso [49]. Although the design of LNA is subjected to keep a minimum NF ( $\leq 2.5$ ), a power consumption of 8 mW is maintained with a moderately high quality factor ( $Q \in [3, 5]$ ) and a drain current of 1 mA excluding the bias reference circuit. Further, it is crucial to maintain a low NF without sacrificing much gain during the design process. Otherwise noise gets transmitted to other blocks of receiver system. In view of this, design problem is transformed into a multi-objective constrained

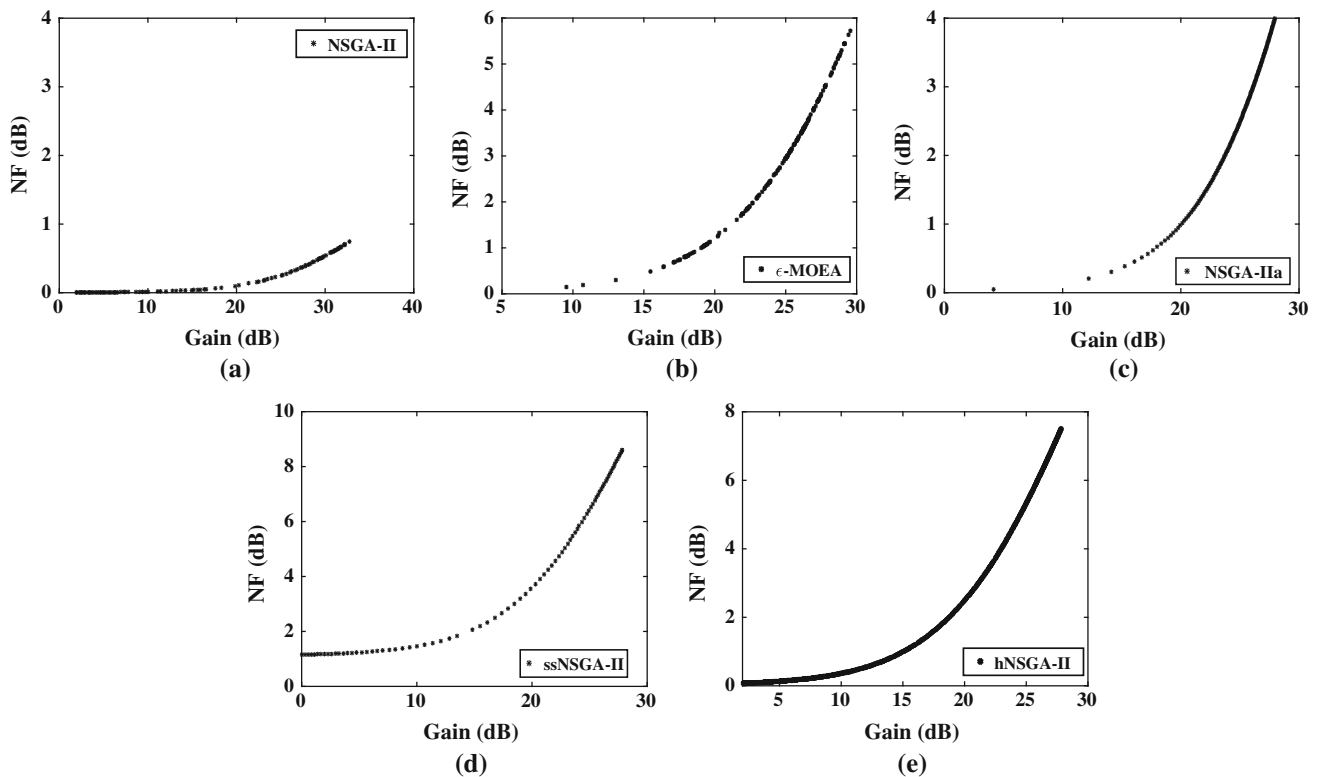


**Fig. 9** Different S-parameters and noise figure of LNA

optimization problem with an objective to minimize NF and to maximize gain<sup>2</sup> ( $S_{21}$  in dB) of LNA. The closed-form expressions of NF and other design constraints are based on the design as reported in [50]. Table 6 summarizes the ranges of variables specified during optimization of LNA. The details of different specifications and constraints during optimization process, and the results obtained after optimization are shown in Table 7.

In order to have a qualitative study of LNA design, simulation results of S-parameters and NF at different frequencies are plotted and shown in Fig. 9. It shows that NF and gain are minimum and maximum respectively at 2.4 GHz center frequency. To start the process of optimization, initial values are assigned to design parameters from first-cut design of LNA. A total of 200 populations are evolved over 500 generations to ensure an optimal PF during LNA optimization. Figure 10 plots the distributions

<sup>2</sup> Here, transducer power gain ( $G_T$ ) is considered as gain, which can be represented as,  $G_T = |S_{21}|^2$  or  $G_T = 20 \log |S_{21}|$  in dB as the source and load impedances are matched to the reference impedance during the design of LNA.



**Fig. 10** Plot of nondominated fronts for gain (max) and noise figure (min) of low noise amplifier by **a** NSGA-II, **b**  $\epsilon$ -MOEA, **c** NSGA-IIa, **d** ssNSGA-II, **e**  $h$ NSGA-II

of final nondominated solutions found in several optimization techniques including proposed  $h$ NSGA-II for LNA optimization. It can be seen from Fig. 10 that the distribution of nondominated solutions in the PF is quiet nonuniform, i.e., solutions are crowded in a corner of the PF rather being distributed in the objective space or the distribution of solutions is not continuous along the PF, for each optimization technique except  $h$ NSGA-II approach. Figure 10(e) represents an optimal PF for minimum NF and maximum gain (S21 parameter) of LNA generated using  $h$ NSGA-II approach. It can be seen from this plot that with increase in gain below 15 dB, NF increases slowly and after that NF starts to increase significantly at high gain values. LNAs having high NF deteriorates the performance of a receiver system significantly by transmitting input signals coupled with large noise to other blocks. Therefore, the feasible region of operation is to operate at a minimum noise figure of 0.67 dB at a moderate gain of 13.01 dB with a power consumption of less than 8 mW.

### 5.3 Performance evaluation

To measure the performance in terms of convergence towards optimal Pareto front and diversity or spread of solutions along Pareto front, *Hypervolume* (HV) metric [23] is evaluated for all peer algorithms including proposed

**Table 8** Median and IQR of the HV metric for analog/RF circuit optimization problems

| Algorithm        | Two-stage opamp       | Low noise amplifier   |
|------------------|-----------------------|-----------------------|
| NSGA-II          | 0.6045 (4.12e−04) [+] | 0.6012 (3.12e−04) [−] |
| $\epsilon$ -MOEA | 0.6073 (4.18e−04) [+] | 0.6210 (1.12e−03) [−] |
| NSGA-IIa         | 0.6101 (3.92e−04) [+] | 0.6273 (4.20e−04) [−] |
| ssNSGA-II        | 0.6602 (2.01e−04) [+] | 0.7225 (1.71e−04) [+] |
| $h$ NSGA-II      | 0.6828 (3.76e−04) [+] | 0.7478 (3.28e−04) [+] |

‘opamp’ denotes operational amplifier

$h$ NSGA-II algorithm. Both median and interquartile range (IQR) values of HV metric obtained after 25 independent runs are listed in Table 8 (IQR is reported within braces).

As HV is a fair measure of the maximum area covered by the nondominated solutions, Pareto fronts having higher HV values represent better quality of optimized solutions in terms of both convergence and diversity of solutions. It can be observed from Table 8 that the proposed  $h$ NSGA-II algorithm achieves superior HV values in both test circuits with statistical confidence (‘+’ symbol denotes a confidence level of 95% [51]).

## 6 Conclusion

This paper presents a multi-objective framework based on *h*NSGA-II and its application in optimization of analog and RF circuits. A new hierarchical scheme in polynomial mutation operation is employed in this framework to produce random generation of improved individuals to enhance the possibility of independent local search in each generation. The stochastic behavior of proposed framework is demonstrated by developing a Markov chain model with bounded ergodicity. Superiority of hierarchical scheme in polynomial mutation operation is shown by comparing the performance with six other mutation operators. Applicability of proposed *h*NSGA-II method is shown by carrying out extensive simulations on several multi-objective test problems, and the effectiveness of this method is verified by generating optimal Pareto fronts for a two-stage operational amplifier and a cascode low noise amplifier with inductive source degeneration. Although simulation results support improvement in the performance of *h*NSGA-II, there is still scope of improvement in both convergence and diversity preservation of Pareto optimal solutions by preserving unique fitness values in each generation. A comprehensive study of hierarchical scheme with real and binary mutation operators can be performed in future to extend the application of proposed method to analyze multi-objective optimization problems having large scale decision variables. Further, the hierarchical scheme can be subjected to parallelization by using efficient parallel strategies over multicore and manycore platforms to reduce the number of function evaluations during optimization process.

## References

- Mandal, P., & Visvanathan, V. (2001). Cmos op-amp sizing using a geometric programming formulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(1), 22–38.
- Martens, E., & Gielen, G. (2008). Classification of analog synthesis tools based on their architecture selection mechanisms. *Integration, the VLSI Journal*, 41(2), 238–252. <https://doi.org/10.1016/j.vlsi.2007.06.001>.
- Michal, J., & Dobes, J. (2007). Electronic circuit design using multiobjective optimization. In *2007 50th Midwest symposium on circuits and systems* (pp. 734–737). IEEE.
- Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A., & Dunlap, F. (1997). Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on Evolutionary Computation*, 1(2), 109–128. <https://doi.org/10.1109/4235.687879>.
- Das, S., Mallipeddi, R., & Maity, D. (2013). Adaptive evolutionary programming with p-best mutation strategy. *Swarm and Evolutionary Computation*, 9, 58–68. <https://doi.org/10.1016/j.swevo.2012.11.002>.
- Fakhfakh, M., Cooren, Y., Sallem, A., Loulou, M., & Siarry, P. (2010). Analog circuit design optimization through the particle swarm optimization technique. *Analog Integrated Circuits and Signal Processing*, 63(1), 71–82.
- Kubař, M., & Jakovenko, J. (2013). A powerful optimization tool for analog integrated circuits design. *Radioengineering*, 22(3), 921.
- Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2), 88–105.
- Cheng, R., Gen, M., & Tsujimura, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms: I. Representation. *Computers & industrial engineering*, 30(4), 983–997.
- Grefenstette, J., Gopal, R., Rosmaita, B., & Van Gucht, D. (1985). Genetic algorithms for the traveling salesman problem. In *Proceedings of the first international conference on genetic algorithms and their applications*, Lawrence Erlbaum, New Jersey (pp. 160–168).
- Changdar, C., Mahapatra, G., & Pal, R. K. (2014). An efficient genetic algorithm for multi-objective solid travelling salesman problem under fuzziness. *Swarm and Evolutionary Computation*, 15, 27–37. <https://doi.org/10.1016/j.swevo.2013.11.001>. <http://www.sciencedirect.com/science/article/pii/S2210650213000679>.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3(2), 95–99.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4), 577–601.
- Miettinen, K. (2012). *Nonlinear multiobjective optimization* (Vol. 12). New York: Springer.
- Suzuki, J. (1995). A markov chain analysis on simple genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(4), 655–659. <https://doi.org/10.1109/21.370197>.
- Deb, K., Mohan, M., & Mishra, S. (2003). A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. *KanGAL Report, 2003002*, 1–18.
- Iosifescu, M. (2014). *Finite Markov processes and their applications*. North Chelmsford: Courier Corporation.
- Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1), 96–101.
- Tweedie, R. L. (1974). *R*-theory for Markov chains on a general state space I: Solidarity properties and *r*-recurrent chains. *The Annals of Probability*, 2(5), 840–864.
- Doob, J. L., & Doob, J. L. (1953). *Stochastic processes* (Vol. 7, No. 2). New York: Wiley.
- Jiang, S., Ong, Y.-S., Zhang, J., & Feng, L. (2014). Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE Transactions on Cybernetics*, 44(12), 2391–2404.
- Fonseca, C. M., Paquete, L., & López-Ibáñez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In: 2006 IEEE international conference on evolutionary computation (pp. 1157–1163). IEEE.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st ed.). Boston, MA: Addison-Wesley Longman Publishing Co., Inc.
- Lee, C.-Y., & Yao, X. (2001). Evolutionary algorithms with adaptive lévy mutations. In *Proceedings of the 2001 congress on evolutionary computation, 2001* (Vol. 1, pp. 568–575). IEEE.

26. Chellapilla, K. (1998). Combining mutation operators in evolutionary programming. *IEEE Transactions on Evolutionary Computation*, 2(3), 91–96.
27. Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2), 82–102. <https://doi.org/10.1109/4235.771163>.
28. Michalewicz, Z., & Janikow, C. Z. (1991). Handling constraints in genetic algorithms. In *ICGA* (pp. 151–157).
29. Bäck, T., & Schwefel, H.-P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1), 1–23.
30. Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2), 284–302.
31. Durillo, J., Nebro, A., Luna, F., & Alba, E. (2009). On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In *5th International conference, EMO 2009. Lecture notes in computer science* (Vol. 5467, pp. 183–197). Berlin: Springer.
32. Rutenbar, R. A., Gielen, G., & Antao, B. (2002). *Computer-aided design of analog integrated circuits and systems*. Piscataway: IEEE Press.
33. Boyd, S. P., Lee, T. H., et al. (2001). Optimal design of a CMOS op-amp via geometric programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(1), 1–21.
34. Harjani, R., Rutenbar, R. A., & Carley, L. R. (1989). Oasys: A framework for analog circuit synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(12), 1247–1266.
35. Ochotta, E. S., Rutenbar, R. A., & Carley, L. R. (1996). Synthesis of high-performance analog circuits in ASTRX/OBLX. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(3), 273–294.
36. Gielen, G., Walscherts, H., & Sansen, W. (1989). Analog circuit design optimization based on symbolic simulation and simulated annealing. In *Solid-state circuits conference, 1989. ESSCIRC '89. Proceedings of the 15th European* (pp. 252–255).
37. Kruiskamp, W., & Leenaerts, D. (1995). Darwin: CMOS opamp synthesis by means of a genetic algorithm. In *Proceedings of the 32nd annual ACM/IEEE design automation conference, DAC '95* (pp. 433–438). New York, NY: ACM .
38. Karaboga, N., Kalinli, A., & Karaboga, D. (2004). Designing digital IIR filters using ant colony optimisation algorithm. *Engineering Applications of Artificial Intelligence*, 17(3), 301–309.
39. Tsai, J.-T., Chou, J.-H., & Liu, T.-K. (2006). Optimal design of digital IIR filters by using hybrid taguchi genetic algorithm. *IEEE Transactions on Industrial Electronics*, 53(3), 867–879.
40. Karaboga, N., & Cetinkaya, B. (2006). Design of digital FIR filters using differential evolution algorithm. *Circuits, Systems, and Signal Processing*, 25(5), 649–660.
41. Gentili, P., Piazza, F., & Uncini, A. (1995). Efficient genetic algorithm design for power-of-two fir filters. In *1995 International conference on acoustics, speech, and signal processing, 1995. ICASSP-95* (Vol. 2, pp. 1268–1271). IEEE.
42. Cadence inc., products: Composer, virtuoso, diva, neocircuit, neocell, ultrasim, ncsim.
43. Analog design automation, inc.
44. del Mar Hershenson, M., Boyd, S. P., & Lee, T. H. (1998). Gpcad: A tool for CMOS op-amp synthesis. In *Proceedings of the 1998 IEEE/ACM international conference on Computer-aided design* (pp. 296–303). ACM.
45. Weber, T. O., & Van Noije, W. A. (2011). Analog design synthesis method using simulated annealing and particle swarm optimization. In *Proceedings of the 24th symposium on Integrated circuits and systems design* (pp. 85–90). ACM.
46. Barros, M., Guilherme, J., & Horta, N. (2010). Analog circuits optimization based on evolutionary computation techniques. *Integration, the VLSI Journal*, 43(1), 136–155.
47. Rabuske, T., & Fernandes, J. (2014). Noise-aware simulation-based sizing and optimization of clocked comparators. *Analog Integrated Circuits and Signal Processing*, 81(3), 723–728.
48. Allen, P. E., & Holberg, D. R. (2002). *CMOS analog circuit design*. Oxford: Oxford University Press.
49. Cadence design systems, inc., virtuoso schematic editor. Available at [https://www.cadence.com/content/dam/cadence-www/global/en\\_us/documents/tools/custom-ic-analog-rf-design/virtuoso-vse-fam-ds.pdf](https://www.cadence.com/content/dam/cadence-www/global/en_us/documents/tools/custom-ic-analog-rf-design/virtuoso-vse-fam-ds.pdf).
50. Andreani, P., & Sjoland, H. (2001). Noise optimization of an inductively degenerated cmos low noise amplifier. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 48(9), 835–841.
51. Nebro, A. J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J. J., & Beham, A. (2008). AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 12(4), 439–457.



**Satyabrata Dash** received the B.Tech. degree in Electronics and Communication Engineering from Biju Patnaik University of Technology, Odisha, India, in 2010. Currently, he is working toward the Ph.D. degree in the Department of Electronics and Electrical Engineering at Indian Institute of Technology Guwahati (IIT Guwahati), India. His current research interests include nature inspired algorithms and its application to VLSI circuit

optimization, analog circuit design and high performance computing. He received the Honorable Mention Award in the 2016 IEEE International Conference on VLSI Design (VLSID), two Best Paper Awards in IEEE International Conference Radioelektronika (RADIOELEKTRONIKA), in years 2015 and 2016.



**Deepak Joshi** received the B.E. degree in Electronics and Communication Engineering from Government Engineering College Bikaner, Rajasthan, India, in 2009. Currently, he is working toward the Ph.D. degree in the Department of Electronics and Electrical Engineering at Indian Institute of Technology Guwahati (IIT Guwahati), India. His current research interests include analog/RF circuit design and analysis. He received a Best Paper Award in IEEE International Conference Radioelektronika (RADIOELEKTRONIKA), in year 2015.

ational Conference Radioelektronika (RADIOELEKTRONIKA), in year 2015.



**Ayushparth Sharma** received the B.Tech. degree in Electronics and Communication Engineering from LNM Institute of Information Technology, Jaipur in the year 2016. He is working at IIT Guwahati as an intern on the development of Multi-objective optimization algorithms for circuit optimization. He also worked as an intern at IIT Kharagpur in 2014 on ‘Synthesis of Low Power High-Performance CMOS Circuits’, sponsored by Government of

India. His research interests involve the development of low-power and high-performance systems emphasizing on algorithm, architecture and circuit explorations.



**Gaurav Trivedi** received the B.E. degree in Electronics from Devi Ahilya University, Indore, India, in 1998. He received the M.Tech. degree in Microelectronics and the Ph.D. degree in the area of VLSI CAD from Indian Institute of Technology Bombay, India in 2000 and in 2007, respectively. He worked as senior member of technical staff with Cadence Design Systems and Berkeley Design Automation for 3 years. Later, he joined Indian Institute of

Technology Bombay, India in 2009 as a post doctoral fellow for

2 years. Currently, he is an Assistant Professor in the Department of Electronics and Electrical engineering at Indian Institute of Technology Guwahati, India. His research interests include VLSI CAD, semiconductor devices, digital and analog circuits design and optimization, high performance computing, computer architecture and algorithms, computational biology and solar photovoltaics.